JUNIPER NETWORKS

Junos® OS

CLI User Guide

Release
14.1

Published: 2014-09-26

Copyright © 2014, Juniper Networks, Inc.

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Junos® OS CLI User Guide
14.1
Copyright © 2014, Juniper Networks, Inc.
All rights reserved.

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at http://www.juniper.net/support/eula.html. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

About the Documentation xv
Documentation and Release Notes xv
Supported Platforms xv
Using the Examples in This Manual xvi
Merging a Full Example xvi
Merging a Snippet xvii
Documentation Conventions xvii
Documentation Feedback xix
Requesting Technical Support xx
Self-Help Online Tools and Resources xx
Opening a Case with JTAC xx

Chapter 1 **Overview 23**

Introducing the Junos OS Command-Line Interface 23
Key Features of the CLI 24
Understanding the Junos OS CLI Modes, Commands, and Statement Hierarchies 25
Junos OS CLI Command Modes 25
CLI Command Hierarchy 26
Configuration Statement Hierarchy 26
Moving Among Hierarchy Levels 27
Other Tools to Configure and Monitor Devices Running Junos OS 28
Commands and Configuration Statements for Junos-FIPS 28

Chapter 2 **Getting Started: A Quick Tour of the CLI 31**

Getting Started with the Junos OS Command-Line Interface 31
Switching Between Junos OS CLI Operational and Configuration Modes 33
Configuring a User Account on a Device Running Junos OS 34
Using the CLI Editor in Configuration Mode 36
Checking the Status of a Device Running Junos OS 38
Configuring a Routing Protocol 40
Shortcut 41
Longer Configuration 41
Making Changes to a Routing Protocol Configuration 43
Rolling Back Junos OS Configuration Changes 46

Chapter 3 **Getting Online Help 49**

Getting Online Help from the Junos OS Command-Line Interface 49
Getting Help About Commands 49
Getting Help About a String in a Statement or Command 50
Getting Help About Configuration Statements 51

Copyright © 2014, Juniper Networks, Inc.

Getting Help About System Log Messages 51
Junos OS CLI Online Help Features 52
Help for Omitted Statements 52
Using CLI Command Completion 52
Using Command Completion in Configuration Mode 53
Displaying Tips About CLI Commands 53
Examples: Using Command Completion in Configuration Mode 53
Examples: Using the Junos OS CLI Command Completion 55
Displaying the Junos OS CLI Command and Word History 56

Chapter 4 Using Configuration Statements to Configure a Device 57

Understanding Junos OS CLI Configuration Mode 58
Configuration Mode Commands 59
Configuration Statements and Identifiers 60
Configuration Statement Hierarchy 62
Entering and Exiting the Junos OS CLI Configuration Mode 64
Forms of the configure Command 66
Using the configure exclusive Command 67
Example: Using the configure Command 68
Modifying the Junos OS Configuration 69
Adding Junos OS Configuration Statements and Identifiers 69
Deleting a Statement from a Junos OS Configuration 71
Example: Deleting a Statement from the Junos OS Configuration 72
Copying a Junos OS Statement in the Configuration 73
Example: Copying a Statement in the Junos OS Configuration 74
Issuing Relative Junos OS Configuration Mode Commands 74
Renaming an Identifier in a Junos OS Configuration 75
Example: Renaming an Identifier in a Junos OS Configuration 75
Inserting a New Identifier in a Junos OS Configuration 75
Example: Inserting a New Identifier in a Junos OS Configuration 76
Example: Using the Wildcard Command with the Range Option 78
Deactivating and Reactivating Statements and Identifiers in a Junos OS Configuration 82
Examples: Deactivating and Reactivating Statements and Identifiers in a Junos OS Configuration 83
Adding Comments in a Junos OS Configuration 84
Example: Including Comments in a Junos OS Configuration 85
Updating the configure private Configuration 86
Displaying the Current Junos OS Configuration 87
Example: Displaying the Current Junos OS Configuration 88
Displaying Additional Information About the Junos OS Configuration 89
Displaying set Commands from the Junos OS Configuration 92
Example: Displaying set Commands from the Configuration 92
Example: Displaying Required set Commands at the Current Hierarchy Level 93
Example: Displaying set Commands with the match Option 93
Displaying Users Currently Editing the Junos OS Configuration 94
Verifying a Junos OS Configuration 95

Copyright © 2014, Juniper Networks, Inc.

Chapter 5 Committing a Junos OS Configuration 97
Junos OS Commit Model for Router or Switch Configuration 97
Committing a Junos OS Configuration 98
Committing a Junos OS Configuration and Exiting Configuration Mode 100
Commit Operation When Multiple Users Configure the Software 101
Activating a Junos OS Configuration but Requiring Confirmation 102
Scheduling a Junos OS Commit Operation 103
Monitoring the Junos OS Commit Process 104
Adding a Comment to Describe the Committed Configuration 105
Backing Up the Committed Configuration on the Alternate Boot Drive 106
Junos OS Batch Commits Overview 107
Aggregation and Error Handling 107
Example: Configuring Batch Commit Server Properties 107
Chapter 6 Managing Configurations 117
Understanding How the Junos OS Configuration Is Stored 117
Comparing Configuration Changes with a Prior Version 118
Returning to the Most Recently Committed Junos OS Configuration 120
Returning to a Previously Committed Junos OS Configuration 120
Returning to a Configuration Prior to the One Most Recently Committed . . 120
Displaying Previous Configurations 121
Comparing Configuration Changes with a Prior Version 122
Creating and Returning to a Rescue Configuration 123
Saving a Configuration to a File 124
Saving a Configuration to a File 125
Additional Details About Specifying Junos OS Statements and Identifiers 126
Specifying Statements 126
Performing CLI Type Checking 128
Loading a Configuration from a File 129
Examples: Loading a Configuration from a File 132
Creating and Returning to a Rescue Configuration 135
Compressing the Current Configuration File 135
Example: Protecting the Junos OS Configuration from Modification or Deletion 137
Synchronizing Routing Engines 144
Configuring Multiple Routing Engines to Synchronize Committed Configurations Automatically 147
Chapter 7 Using Operational Commands to Monitor a Device 149
Overview of Junos OS CLI Operational Mode Commands 149
CLI Command Categories 149
Commonly Used Operational Mode Commands 151
Junos OS Operational Mode Commands That Combine Other Commands 152
Understanding the Brief, Detail, Extensive, and Terse Options of Junos OS Operational Commands 153

Copyright © 2014, Juniper Networks, Inc.

Controlling the Scope of an Operational Mode Command 154
Operational Mode Commands on a TX Matrix Router or TX Matrix Plus Router 155
Examples of Routing Matrix Command Options 155
Monitoring Who Uses the Junos OS CLI 157
Interface Naming Conventions Used in the Junos OS Operational Commands 158
Physical Part of an Interface Name 158
Logical Part of an Interface Name 158
Channel Identifier Part of an Interface Name 159
Viewing Files and Directories on a Device Running Junos OS 159
Directories on the Router or Switch 159
Listing Files and Directories 160
Specifying Filenames and URLs 162
Displaying Junos OS Information 163
Managing Programs and Processes Using Junos OS Operational Mode Commands 165
Showing Software Processes 166
Restarting a Junos OS Process 167
Stopping Junos OS 168
Rebooting Junos OS 169
Using the Junos OS CLI Comment Character # for Operational Mode Commands 170
Example: Using Comments in Junos OS Operational Mode Commands 170

Chapter 8 **Filtering Command Output 173**

Using the Pipe (|) Symbol to Filter Junos OS Command Output 173
Using Regular Expressions with the Pipe (|) Symbol to Filter Junos OS Command Output 174
Filtering Operational Mode Command Output in a QFabric System 175
Pipe (|) Filter Functions in the Junos OS Command-Line Interface 176
Comparing Configurations 176
Counting the Number of Lines of Output 178
Displaying Output in XML Tag Format 178
Displaying the RPC tags for a Command 178
Ignoring Output That Does Not Match a Regular Expression 179
Displaying Output from the First Match of a Regular Expression 179
Retaining Output After the Last Screen 179
Displaying Output Beginning with the Last Entries 180
Displaying Output That Matches a Regular Expression 180
Preventing Output from Being Paginated 180
Sending Command Output to Other Users 181
Resolving IP Addresses 181
Saving Output to a File 181
Appending Output to a File 182
Displaying Output on Screen and Writing to a File 182
Trimming Output by Specifying the Starting Column 182
Refreshing the Output of a Command 183

Copyright © 2014, Juniper Networks, Inc.

Chapter 9 Using Shortcuts, Wildcards, and Regular Expressions in the CLI 185
Using Keyboard Sequences to Move Around and Edit the Junos OS CLI 185
Using Wildcard Characters in Interface Names 187
Common Regular Expressions to Use with the replace Command 188
Using Global Replace in a Junos OS Configuration 189
Example: Using Global Replace in a Junos OS Configuration—Using the \n Back Reference 190
Example: Using Global Replace in a Junos OS Configuration—Replacing an Interface Name 191
Example: Using Global Replace in a Junos OS Configuration—Using the upto Option 192
Using Regular Expressions to Delete Related Items from a Junos OS cConfiguration 193
Chapter 10 Using Configuration Groups to Quickly Configure Devices 197
Understanding Junos OS Configuration Groups 198
Configuration Groups Overview 198
Inheritance Model 198
Configuring Configuration Groups 198
Creating a Junos OS Configuration Group 199
Applying a Junos OS Configuration Group 201
Example: Configuring and Applying Junos OS Configuration Groups 202
Example: Creating and Applying Configuration Groups on a TX Matrix Router . . 203
Disabling Inheritance of a Junos OS Configuration Group 204
Using Wildcards with Configuration Groups 206
Example: Configuring Sets of Statements with Configuration Groups 209
Example: Configuring Interfaces Using Junos OS Configuration Groups 210
Example: Configuring a Consistent IP Address for the Management Interface . . 212
Example: Configuring Peer Entities 214
Establishing Regional Configurations 215
Selecting Wildcard Names 217
Example: Referencing the Preset Statement From the Junos OS defaults Group 218
Example: Viewing Default Statements That Have Been Applied to the Configuration 219
Using Conditions to Apply Configuration Groups Overview 219
Example: Configuring Conditions for Applying Configuration Groups 220
Improving Commit Time When Using Configuration Groups 222
Example: Improving Commit Time When Using Configuration Groups 223
Using Junos OS Defaults Groups 224
Set Up Routing Engine Configuration Groups 225
Chapter 11 Controlling the CLI Environment 229
Controlling the Junos OS CLI Environment 229
Setting the Terminal Type 230
Setting the CLI Prompt 230
Setting the CLI Directory 230
Setting the CLI Timestamp 230
Setting the Idle Timeout 230

Copyright © 2014, Juniper Networks, Inc.

Setting the CLI to Prompt After a Software Upgrade 230
Setting Command Completion 231
Displaying CLI Settings 231
Setting the Junos OS CLI Screen Length and Width 231
Setting the Screen Length 231
Setting the Screen Width 232
Understanding the Screen Length and Width Settings 232
Example: Controlling the CLI Environment 232
Example: Enabling Configuration Breadcrumbs 233

Chapter 12 Junos OS Configuration Statements and Commands 235

apply-groups 236
apply-groups-except 237
activate 238
annotate 239
commit 240
commit-interval (Batch Commits) 245
configuration-breadcrumbs 245
copy 246
days-to-keep-error-logs (Batch Commits) 246
deactivate 247
delete 248
edit 249
exit 250
groups 251
help 253
insert 254
load 255
maximum-aggregate-pool (Batch Commits) 256
maximum-entries (Batch Commits) 257
protect 258
quit 259
rename 260
replace 261
rollback 262
run 263
save 264
server (Batch Commits) 265
set 266
show 267
show configuration 268
show | display inheritance 271
show | display omit 272
show | display set 273
show | display set relative 274
show groups junos-defaults 275
status 276
top 277
traceoptions (Batch Commits) 278

Copyright © 2014, Juniper Networks, Inc.

unprotect . . . 279
up . . . 280
update . . . 281
when . . . 282
wildcard delete . . . 283

Chapter 13 Junos OS CLI Environment Commands . . . 285

set cli complete-on-space . . . 286
set cli directory . . . 287
set cli idle-timeout . . . 288
set cli prompt . . . 289
set cli restart-on-upgrade . . . 290
set cli screen-length . . . 291
set cli screen-width . . . 292
set cli terminal . . . 293
set cli timestamp . . . 294
set date . . . 295
show cli . . . 296
show cli . . . 298
show cli authorization . . . 299
show cli directory . . . 301
show cli history . . . 302

Chapter 14 Junos OS CLI Operational Mode Commands . . . 303

configure . . . 304
file . . . 306
help . . . 308
| (pipe) . . . 309
request . . . 311
request system commit server pause . . . 313
request system commit server queue cleanup . . . 314
request system commit server start . . . 315
restart . . . 316
set . . . 327
show system commit server queue . . . 328
show system commit server status . . . 332

Chapter 15 Index . . . 333

Index . . . 335

Copyright © 2014, Juniper Networks, Inc.

Copyright © 2014, Juniper Networks, Inc.

List of Figures

Chapter 1 Overview 23

Figure 1: Monitoring and Configuring Routers 24
Figure 2: Committing a Configuration 26
Figure 3: Configuration Statement Hierarchy Example 27

Chapter 4 Using Configuration Statements to Configure a Device 57

Figure 4: Configuration Mode Hierarchy of Statements 62

Chapter 5 Committing a Junos OS Configuration 97

Figure 5: Confirm a Configuration 103

Chapter 6 Managing Configurations 117

Figure 6: Overriding the Current Configuration 132
Figure 7: Using the replace Option 133
Figure 8: Using the merge Option 133
Figure 9: Using a Patch File 134
Figure 10: Using the set Option 134

Chapter 7 Using Operational Commands to Monitor a Device 149

Figure 11: Commands That Combine Other Commands 153
Figure 12: Command Output Options 154
Figure 13: Restarting a Process 168

Chapter 9 Using Shortcuts, Wildcards, and Regular Expressions in the CLI 185

Figure 14: Replacement by Object 192

Copyright © 2014, Juniper Networks, Inc.

Copyright © 2014, Juniper Networks, Inc.

List of Tables

About the Documentation xv

Table 1: Notice Icons xviii
Table 2: Text and Syntax Conventions xviii

Chapter 1 Overview 23

Table 3: CLI Configuration Mode Navigation Commands 27

Chapter 4 Using Configuration Statements to Configure a Device 57

Table 4: Summary of Configuration Mode Commands 59
Table 5: Configuration Mode Top-Level Statements 61
Table 6: Forms of the configure Command 66

Chapter 6 Managing Configurations 117

Table 7: CLI Configuration Input Types 128

Chapter 7 Using Operational Commands to Monitor a Device 149

Table 8: Commonly Used Operational Mode Commands 151
Table 9: Directories on the Router 160
Table 10: show system process extensive Command Output Fields 167

Chapter 8 Filtering Command Output 173

Table 11: Common Regular Expression Operators in Operational Mode Commands 174

Chapter 9 Using Shortcuts, Wildcards, and Regular Expressions in the CLI 185

Table 12: CLI Keyboard Sequences 186
Table 13: Wildcard Characters for Specifying Interface Names 187
Table 14: Common Regular Expressions to Use with the replace Command . . . 188
Table 15: Replacement Examples 189

Chapter 13 Junos OS CLI Environment Commands 285

Table 16: show cli Output Fields 296
Table 17: show cli directory Output Fields 301

Copyright © 2014, Juniper Networks, Inc.

Copyright © 2014, Juniper Networks, Inc.

About the Documentation

- Documentation and Release Notes on page xv
- Supported Platforms on page xv
- Using the Examples in This Manual on page xvi
- Documentation Conventions on page xvii
- Documentation Feedback on page xix
- Requesting Technical Support on page xx

Documentation and Release Notes

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at http://www.juniper.net/techpubs/.

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at http://www.juniper.net/books.

Supported Platforms

For the features described in this document, the following platforms are supported:

- ACX Series
- M Series
- MX Series
- T Series
- J Series
- PTX Series

Copyright © 2014, Juniper Networks, Inc.

Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the **load merge** command.

If the example configuration does not start at the top level of the hierarchy, the example is a *snippet*. In this case, use the **load merge relative** command. These procedures are described in the following sections.

Merging a Full Example

To merge a full example, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration example into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following configuration to a file and name the file **ex-script.conf**. Copy the **ex-script.conf** file to the **/var/tmp** directory on your routing platform.

   ```
   system {
     scripts {
       commit {
         file ex-script.xsl;
       }
     }
   }
   interfaces {
     fxp0 {
       disable;
       unit 0 {
         family inet {
           address 10.0.0.1/24;
         }
       }
     }
   }
   ```

2. Merge the contents of the file into your routing platform configuration by issuing the **load merge** configuration mode command:

   ```
   [edit]
   user@host# load merge /var/tmp/ex-script.conf
   load complete
   ```


Copyright © 2014, Juniper Networks, Inc.

Merging a Snippet

To merge a snippet, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration snippet into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following snippet to a file and name the file **ex-script-snippet.conf**. Copy the **ex-script-snippet.conf** file to the **/var/tmp** directory on your routing platform.

   ```
   commit {
     file ex-script-snippet.xsl; }
   ```

2. Move to the hierarchy level that is relevant for this snippet by issuing the following configuration mode command:

   ```
   [edit]
   user@host# edit system scripts
   [edit system scripts]
   ```

3. Merge the contents of the file into your routing platform configuration by issuing the **load merge relative** configuration mode command:

   ```
   [edit system scripts]
   user@host# load merge relative /var/tmp/ex-script-snippet.conf
   load complete
   ```

For more information about the **load** command, see the *CLI User Guide*.

Documentation Conventions

Table 1 on page xviii defines notice icons used in this guide.

Copyright © 2014, Juniper Networks, Inc.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.
	Tip	Indicates helpful information.
	Best practice	Alerts you to a recommended use or implementation.

Table 2 on page xviii defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces or emphasizes important new terms. • Identifies guide names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS CLI User Guide* • RFC 1997, *BGP Communities Attribute*
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*

Copyright © 2014, Juniper Networks, Inc.

Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Encloses optional keywords or variables.	**stub <default-metric *metric*>;**
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** **(*string1* | *string2* | *string3*)**
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Encloses a variable for which you can substitute one or more values.	**community name members [*community-ids*]**

Indention and braces ({ })	Identifies a level in the configuration hierarchy.	[edit]
		routing-options {
		  static {
		  route default {
		  nexthop *address*;
		  retain;
		  }
		  }
		}

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		
Bold text like this	Represents graphical user interface (GUI) items you click or select.	• In the Logical Interfaces box, select **All Interfaces**. • To cancel the configuration, click **Cancel**.
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select **Protocols>Ospf**.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can provide feedback by using either of the following methods:

- Online feedback rating system—On any page at the Juniper Networks Technical Documentation site at http://www.juniper.net/techpubs/index.html, simply click the stars to rate the content, and use the pop-up form to provide us with information about your experience. Alternately, you can use the online feedback form at https://www.juniper.net/cgi-bin/docbugreport/.

Copyright © 2014, Juniper Networks, Inc.

- E-mail—Send your comments to techpubs-comments@juniper.net. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit http://www.juniper.net/support/warranty/.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/
- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: http://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: https://tools.juniper.net/SerialNumberEntitlementSearch/

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at http://www.juniper.net/cm/.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

Copyright © 2014, Juniper Networks, Inc.

For international or direct-dial options in countries without toll-free numbers, see http://www.juniper.net/support/requesting-support.html.

Copyright © 2014, Juniper Networks, Inc.

Copyright © 2014, Juniper Networks, Inc.

CHAPTER 1

Overview

- Introducing the Junos OS Command-Line Interface on page 23
- Understanding the Junos OS CLI Modes, Commands, and Statement Hierarchies on page 25
- Other Tools to Configure and Monitor Devices Running Junos OS on page 28
- Commands and Configuration Statements for Junos-FIPS on page 28

Introducing the Junos OS Command-Line Interface

The Junos® operating system (Junos OS) command-line interface (CLI) is the software interface you use to access a device running Junos OS—whether from the console or through a network connection.

The Junos OS CLI is a Juniper Networks-specific command shell that runs on top of a FreeBSD UNIX-based operating system kernel. By leveraging industry-standard tools and utilities, the CLI provides a powerful set of commands that you can use to monitor and configure devices running Junos OS (see Figure 1 on page 24).

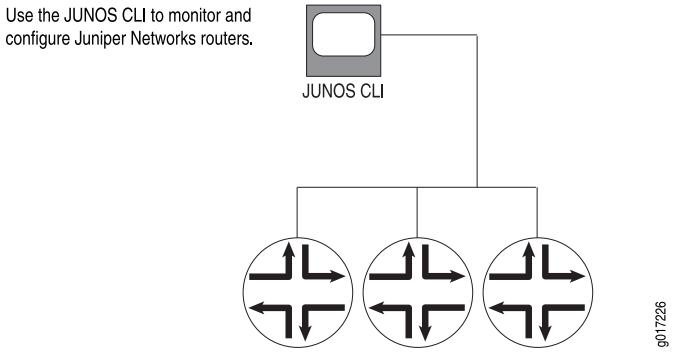
The Junos OS CLI has two modes:

- Operational mode—This mode displays the current status of the device. In operational mode, you enter commands to monitor and troubleshoot the Junos OS, devices, and network connectivity.
- Configuration mode—This mode enables you to configure the device. A configuration is stored as a hierarchy of configuration statements. In this mode, you enter statements to configure all properties of the device, including interfaces, general routing information, routing protocols, user access, and several system and hardware properties.

  When you enter configuration mode, you are actually viewing and changing a file called the *candidate configuration*. The candidate configuration file enables you to make configuration changes without causing operational changes to the current operating configuration, called the *active configuration*. The router or switch does not implement the changes you added to the candidate configuration file until you commit them, which activates the configuration on the device. Candidate configurations enable you to alter your configuration without causing potential damage to your current network operations.

Copyright © 2014, Juniper Networks, Inc.

Figure 1: Monitoring and Configuring Routers



Key Features of the CLI

The Junos OS CLI commands and statements follow a hierarchal organization and have a regular syntax. The Junos OS CLI provides the following features to simplify CLI use:

- Consistent command names—Commands that provide the same type of function have the same name, regardless of the portion of the software on which they are operating. For example, all **show** commands display software information and statistics, and all **clear** commands erase various types of system information.
- Lists and short descriptions of available commands—Information about available commands is provided at each level of the CLI command hierarchy. If you type a question mark (**?**) at any level, you see a list of the available commands along with a short description of each command. This means that if you already are familiar with the Junos OS or with other routing software, you can use many of the CLI commands without referring to the documentation.
- Command completion—Command completion for command names (keywords) and for command options is available at each level of the hierarchy. To complete a command or option that you have partially typed, press the Tab key or the Spacebar. If the partially typed letters begin a string that uniquely identifies a command, the complete command name appears. Otherwise, a beep indicates that you have entered an ambiguous command, and the possible completions are displayed. Completion also applies to other strings, such as filenames, interface names, usernames, and configuration statements.

  If you have typed the mandatory arguments for executing a command in the operational or configuration mode the CLI displays **<[Enter]>** as one of the choices when you type a question mark (**?**). This indicates that you have entered the mandatory arguments and can execute the command at that level without specifying any further options. Likewise, the CLI also displays **<[Enter]>** when you have reached a specific hierarchy level in the configuration mode and do not have to enter any more mandatory arguments or statements.
- Industry-standard technology—With FreeBSD UNIX as the kernel, a variety of UNIX utilities are available on the Junos OS CLI. For example, you can:
  - Use regular expression matching to locate and replace values and identifiers in a configuration, filter command output, or examine log file entries.

Copyright © 2014, Juniper Networks, Inc.

- Use Emacs-based key sequences to move around on a command line and scroll through the recently executed commands and command output.
- Store and archive Junos OS device files on a UNIX-based file system.
  - Use standard UNIX conventions to specify filenames and paths.
  - Exit from the CLI environment and create a UNIX C shell or Bourne shell to navigate the file system, manage router processes, and so on.

Related Documentation

- Understanding the Junos OS CLI Modes, Commands, and Statement Hierarchies on page 25
- Getting Started with the Junos OS Command-Line Interface on page 31
- Other Tools to Configure and Monitor Devices Running Junos OS on page 28
- Commands and Configuration Statements for Junos-FIPS on page 28

Understanding the Junos OS CLI Modes, Commands, and Statement Hierarchies

The Junos OS command-line interface (CLI) commands and statements are organized under two command modes and various hierarchies. The following sections provide you an overview of the Junos OS CLI command modes and commands and statements hierarchies:

- Junos OS CLI Command Modes on page 25
- CLI Command Hierarchy on page 26
- Configuration Statement Hierarchy on page 26
- Moving Among Hierarchy Levels on page 27

Junos OS CLI Command Modes

The Junos OS CLI has two modes:

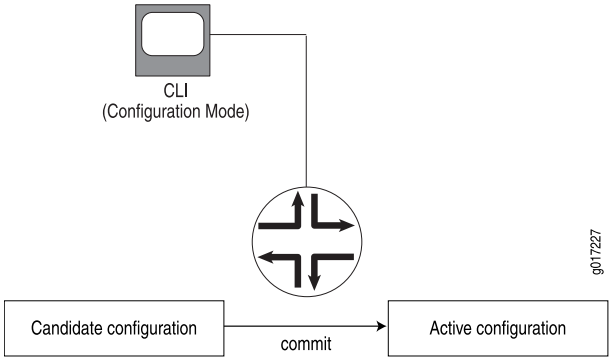
- Operational mode—This mode displays the current status of the device. In operational mode, you enter commands to monitor and troubleshoot the Junos OS, devices, and network connectivity.
- Configuration mode—A configuration for a device running on Junos OS is stored as a hierarchy of statements. In configuration mode, you enter these statements to define all properties of the Junos OS, including interfaces, general routing information, routing protocols, user access, and several system and hardware properties.

When you enter configuration mode, you are actually viewing and changing a file called the *candidate configuration*. The candidate configuration file enables you to make configuration changes without causing operational changes to the current operating configuration, called the *active configuration*. The router or switch does not implement the changes you added to the candidate configuration file until you commit them, which activates the configuration on the router or switch (see Figure 2 on page 26). Candidate

Copyright © 2014, Juniper Networks, Inc.

configurations enable you to alter your configuration without causing potential damage to your current network operations.

Figure 2: Committing a Configuration



CLI Command Hierarchy

CLI commands are organized in a hierarchy. Commands that perform a similar function are grouped together under the same level of the hierarchy. For example, all commands that display information about the system and the system software are grouped under the **show system** command, and all commands that display information about the routing table are grouped under the **show route** command.

To execute a command, you enter the full command name, starting at the top level of the hierarchy. For example, to display a brief view of the routes in the routing table, use the command **show route brief**.

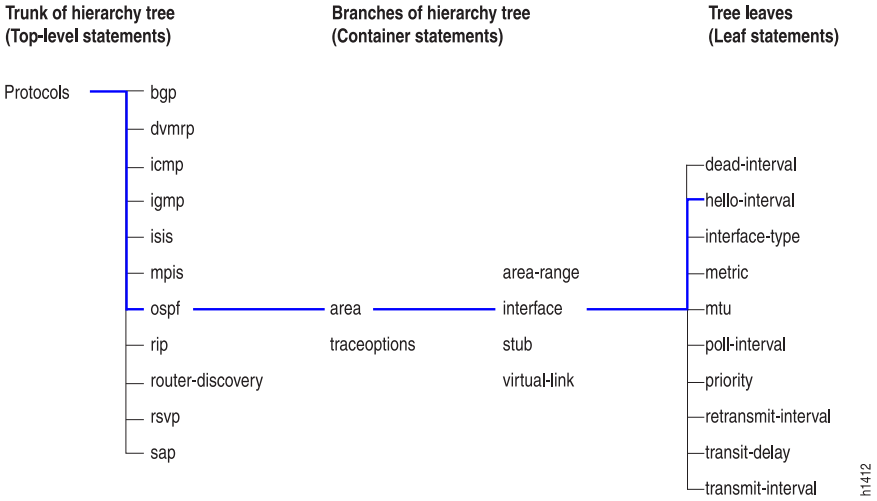
Configuration Statement Hierarchy

The configuration statement hierarchy has two types of statements: *container statements*, which are statements that contain other statements, and *leaf statements*, which do not contain other statements. All of the container and leaf statements together form the *configuration hierarchy*.

Figure 3 on page 27 illustrates a part of the hierarchy tree. The **protocols** statement is a top-level statement at the trunk of the configuration tree. The **ospf**, **area**, and **interface** statements are all subordinate container statements of a higher statement (they are branches of the hierarchy tree), and the **hello-interval** statement is a leaf on the tree.

Copyright © 2014, Juniper Networks, Inc.

Figure 3: Configuration Statement Hierarchy Example



Moving Among Hierarchy Levels

You can use the CLI commands in Table 3 on page 27 to navigate the levels of the configuration statement hierarchy.

Table 3: CLI Configuration Mode Navigation Commands

Command	Description
edit ***hierarchy-level***	Moves to an existing configuration statement hierarchy or creates a hierarchy and moves to that level.
exit	Moves up the hierarchy to the previous level where you were working. This command is, in effect, the opposite of the **edit** command. Alternatively, you can use the **quit** command. The **exit** and **quit** commands are interchangeable.
up	Moves up the hierarchy one level at a time.
top	Moves directly to the top level of the hierarchy.

Related Documentation

- Introducing the Junos OS Command-Line Interface on page 23
- Getting Started with the Junos OS Command-Line Interface on page 31

Copyright © 2014, Juniper Networks, Inc.

Other Tools to Configure and Monitor Devices Running Junos OS

Apart from the command-line interface, Junos OS also supports the following applications, scripts, and utilities that enable you to configure and monitor devices running Junos OS:

- J-Web graphical user interface (GUI)—Allows you to monitor, configure, troubleshoot, and manage the router on a client by means of a Web browser with Hypertext Transfer Protocol (HTTP) or HTTP over Secure Sockets Layer (HTTPS) enabled. For more information, see the *J-Web Interface User Guide*.
- Junos XML management protocol—Application programmers can use the Junos XML management protocol to monitor and configure Juniper Networks routers. Juniper Networks provides a Perl module with the API to help you more quickly and easily develop custom Perl scripts for configuring and monitoring routers. For more information, see the *Junos XML Management Protocol Developer Guide*.
- NETCONF Application Programming Interface (API)—Application programmers can also use the NETCONF XML management protocol to monitor and configure Juniper Networks routers. For more information, see the *NETCONF XML Management Protocol Developer Guide*.
- Junos OS commit scripts and self-diagnosis features—You can define scripts to enforce custom configuration rules, use commit script macros to provide simplified aliases for frequently used configuration statements, and configure diagnostic event policies and actions associated with each policy. For more information, see the *Junos OS Automation Library*.
- Management Information Bases (MIBs)—You can use enterprise-specific and standard MIBS to retrieve information about the hardware and software components on a Juniper Networks router. For more information about MIBs, see the *Network Management Administration Guide for Routing Devices*.

Related Documentation

- Introducing the Junos OS Command-Line Interface on page 23
- Getting Started with the Junos OS Command-Line Interface on page 31
- Commands and Configuration Statements for Junos-FIPS on page 28

Commands and Configuration Statements for Junos-FIPS

Junos-FIPS enables you to configure a network of Juniper Networks routers in a Federal Information Processing Standards (FIPS) 140-2 environment.

The Junos-FIPS software environment requires the installation of FIPS software by a crypto officer. In Junos-FIPS, some Junos OS commands and statements have restrictions and some additional configuration statements are available. For more information, see the *Secure Configuration Guide for Common Criteria and Junos-FIPS*.

Related Documentation

- *Junos Secure Configuration Guide for Common Criteria and Junos-FIPS*
- *IPsec System Requirements for Junos-FIPS*

Copyright © 2014, Juniper Networks, Inc.

- *Configuring Internal IPsec for Junos-FIPS*

Copyright © 2014, Juniper Networks, Inc.

Copyright © 2014, Juniper Networks, Inc.

CHAPTER 2

Getting Started: A Quick Tour of the CLI

- Getting Started with the Junos OS Command-Line Interface on page 31
- Switching Between Junos OS CLI Operational and Configuration Modes on page 33
- Configuring a User Account on a Device Running Junos OS on page 34
- Using the CLI Editor in Configuration Mode on page 36
- Checking the Status of a Device Running Junos OS on page 38
- Configuring a Routing Protocol on page 40
- Rolling Back Junos OS Configuration Changes on page 46

Getting Started with the Junos OS Command-Line Interface

As an introduction to the Junos OS command-line interface (CLI), this topic provides instructions for simple steps you take after installing Junos OS on the device. It shows you how to start the CLI, view the command hierarchy, and make small configuration changes. The related topics listed at the end of this topic provide you more detailed information about using the CLI.

NOTE:

- The instructions and examples in this topic are based on sample M Series and T Series routers. You can use them as a guideline for entering commands on your devices running Junos OS.
- Before you begin, make sure your device hardware is set up and Junos OS is installed. You must have a direct console connection to the device or network access using SSH or Telnet. If your device is not set up, follow the installation instructions provided with the device before proceeding.

To log in to a router and start the CLI:

1. Log in as **root**.

   The root login account has superuser privileges, with access to all commands and statements.

2. Start the CLI:

   ```
   root# cli
   ```

Copyright © 2014, Juniper Networks, Inc.

```
root@>
```

The > command prompt shows you are in operational mode. Later, when you enter configuration mode, the prompt will change to #.

NOTE: If you are using the root account for the first time on the device, remember that the device ships with no password required for root, but the first time you commit a configuration with Junos OS Release 7.6 or later, you must set a root password. Root access is not allowed over a telnet session. To enable root access over an SSH connection, you must configure the **system ?services ssh root-login allow** statement.

The CLI includes several ways to get help about commands. This section shows some examples of how to get help:

1. Type **?** to show the top-level commands available in operational mode.

```
root@> ?
Possible completions:
    clear           Clear information in the system
    configure       Manipulate software configuration information
    diagnose        Invoke diagnose script
    file            Perform file operations
    help            Provide help information
    monitor         Show real-time debugging information
    mtrace          Trace multicast path from source to receiver
    ping            Ping remote target
    quit            Exit the management session
    request         Make system-level requests
    restart         Restart software process
    set             Set CLI properties, date/time, craft interface message
    show            Show system information
    ssh             Start secure shell on another host
    start           Start shell
    telnet          Telnet to another host
    test            Perform diagnostic debugging
    traceroute      Trace route to remote host
```

2. Type **file ?** to show all possible completions for the **file** command.

```
root@> file ?

Possible completions:
    <[Enter]>           Execute this command
    archive             Archives files from the system
    checksum            Calculate file checksum
    compare             Compare files
    copy                Copy files (local or remote)
    delete              Delete files from the system
    list                List file information
    rename              Rename files
    show                Show file contents
    source-address      Local address to use in originating the connection
    |                   Pipe through a command
```

3. Type **file archive ?** to show all possible completions for the **file archive** command.

```
root@> file archive ?
```


Copyright © 2014, Juniper Networks, Inc.

```
Possible completions:
    compress             Compresses the archived file using GNU gzip (.tgz)
    destination          Name of created archive (URL, local, remote, or floppy)
    source               Path of directory to archive
```

Related Documentation

- Getting Online Help from the Junos OS Command-Line Interface on page 49
- Switching Between Junos OS CLI Operational and Configuration Modes on page 33
- Checking the Status of a Device Running Junos OS on page 38
- Configuring a User Account on a Device Running Junos OS on page 34
- Configuring a Routing Protocol on page 40
- Examples: Using the Junos OS CLI Command Completion on page 55

Switching Between Junos OS CLI Operational and Configuration Modes

When you monitor and configure a device running Junos OS, you may need to switch between operational mode and configuration mode. When you change to configuration mode, the command prompt also changes. The operational mode prompt is a right angle bracket (**>**) and the configuration mode prompt is a pound sign (**#**).

To switch between operational mode and configuration mode:

1. When you log in to the router and type the **cli** command, you are automatically in operational mode:

   ```
   ---  JUNOS 9.2B1.8 built 2008-05-09 23:41:29 UTC
   % cli
   user@host>
   ```

2. To enter configuration mode, type the **configure** command or the **edit** command from the CLI operation mode. For example:

   ```
   user@host> configure
   Entering configuration mode

   [edit]
   user@host#
   ```

   The CLI prompt changes from **user@host>** to **user@host#** and a banner appears to indicate the hierarchy level.

3. You can return to operational mode in one of the following ways:

   - To commit the configuration and exit:

     ```
     [edit]
     user@host# commit and-quit
     commit complete
     Exiting configuration mode
     user@host>
     ```

   - To exit without committing:

     ```
     [edit]
     ```

Copyright © 2014, Juniper Networks, Inc.

```
user@host# exit
Exiting configuration mode
user@host>
```

When you exit configuration mode, the CLI prompt changes from **user@host#** to **user@host>** and the banner no longer appears. You can enter or exit configuration mode as many times as you wish without committing your changes.

4. To display the output of an operational mode command, such as **show**, while in configuration mode, issue the **run** configuration mode command and then specify the operational mode command:

```
[edit]
user@host# run operational-mode-command
```

For example, to display the currently set priority value of the Virtual Router Redundancy Protocol (VRRP) primary router while you are modifying the VRRP configuration for a backup router:

```
[edit interfaces xe-4/2/0 unit 0 family inet vrrp-group 27]
user@host# show
virtual-address [ 192.168.1.15 ];
[edit interfaces xe-4/2/0 unit 0 family inet vrrp-group 27]
user@host# run show vrrp detail
Physical interface: xe-5/2/0, Unit: 0, Address: 192.168.29.10/24
Interface state: up, Group: 10, State: backup
Priority: 190, Advertisement interval: 3, Authentication type: simple
Preempt: yes, VIP count: 1, VIP: 192.168.29.55
Dead timer: 8.326, Master priority: 201, Master router: 192.168.29.254
[edit interfaces xe-4/2/0 unit 0 family inet vrrp-group 27]
user@host# set priority ...
```

Related Documentation

- Understanding the Junos OS CLI Modes, Commands, and Statement Hierarchies on page 25
- Getting Online Help from the Junos OS Command-Line Interface on page 49
- Configuring a User Account on a Device Running Junos OS on page 34

Configuring a User Account on a Device Running Junos OS

This topic describes how to log on to a device running Junos OS using a root account and configure a new user account. You can configure an account for your own use or create a test account.

To configure a new user account on the device:

1. Log in as root and enter configuration mode:

```
root@host> configure
[edit]
root@host#
```

The prompt in brackets (**[edit]**), also known as a *banner*, shows that you are in configuration edit mode at the top of the hierarchy.

Copyright © 2014, Juniper Networks, Inc.

2. Change to the **[edit system login]** section of the configuration:

   ```
   [edit]
   root@host# edit system login
   [edit system login]
   root@host#
   ```

   The prompt in brackets changes to **[edit system login]** to show that you are at a new level in the hierarchy.

3. Now add a new user account:

   ```
   [edit system login]
   root@host# edit user nchen
   ```

   This example adds an account **nchen** (for Nathan Chen).

   NOTE: In Junos OS Release 12.2 and later, user account names can contain a period (.) in the name. For example, you can have a user account named nathan.chen. However, the username cannot begin or end with a period.

4. Configure a full name for the account. If the name includes spaces, enclose the entire name in quotation marks (" "):

   ```
   [edit system login user nchen]
   root@host# set full-name "Nathan Chen"
   ```

5. Configure an account class. The account class sets the user access privileges for the account:

   ```
   [edit system login user nchen]
   root@host# set class super-user
   ```

6. Configure an authentication method and password for the account:

   ```
   [edit system login user nchen
   root@host# set authentication plain-text-password
   New password:
   Retype new password:
   ```

   When the new password prompt appears, enter a clear-text password that the system can encrypt, and then confirm the new password.

7. Commit the configuration:

   ```
   [edit system login user nchen]
   root@host# commit
   commit complete
   ```

   Configuration changes are not activated until you commit the configuration. If the commit is successful, a **commit complete** message appears.

8. Return to the top level of the configuration, and then exit:

   ```
   [edit system login user nchen]
   root@host# top
   [edit]
   root@host# exit
   Exiting configuration mode
   ```

Copyright © 2014, Juniper Networks, Inc.

9. Log out of the device:

   ```
   root@host> exit
   % logout Connection closed.
   ```

10. To test your changes, log back in with the user account and password you just configured:

    ```
    login: nchen
    Password: password
    --- Junos 8.3-R1.1 built 2005-12-15 22:42:19 UTC
    nchen@host>
    ```

    When you log in, you should see the new username at the command prompt.

You have successfully used the CLI to view the device status and perform a simple configuration change. See the related topics listed in this section for more information about the Junos OS CLI features.

NOTE: For complete information about the commands to issue to configure your device, including examples, see the Junos OS configuration guides.

Related Documentation

- Getting Started with the Junos OS Command-Line Interface on page 31
- Getting Online Help from the Junos OS Command-Line Interface on page 49
- Displaying the Junos OS CLI Command and Word History on page 56
- Configuring a Routing Protocol on page 40

Using the CLI Editor in Configuration Mode

This topic describes some of the basic commands that you must use to enter configuration mode in the command-line interface (CLI) editor, navigate through the configuration hierarchy, get help, and commit or revert the changes that you make during the configuration session.

Task	Command/Statement	Example
Edit Your Configuration		
Enter configuration mode. When you first log in to the device, the device is in operational mode. You must explicitly enter configuration mode. When you do, the CLI prompt changes from **user@host>** to **user@host#** and the hierarchy level appears in square brackets.	**configure**	user@host> **configure** [edit] user@host#

Copyright © 2014, Juniper Networks, Inc.

Task	Command/Statement	Example
Create a statement hierarchy. You can use the **edit** command to simultaneously create a hierarchy and move to that new level in the hierarchy. You cannot use the **edit** command to change the value of identifiers.	edit *hierarchy-level value*	[edit] user@host# **edit security zones security-zone myzone** [edit security zones security-zone myzone] user@host#
Create a statement hierarchy and set identifier values. The **set** command is similar to **edit** except that your current level in the hierarchy does not change.	set *hierarchy-level value*	[edit] user@host# **set security zones security-zone myzone** [edit] user@host#
Navigate the Hierarchy		
Navigate down to an existing hierarchy level.	edit *hierarchy-level*	[edit] user@host# **edit security zones** [edit security zones] user@host#
Navigate up one level in the hierarchy.	up	[edit security zones] user@host# **up** [edit security] user@host#
Navigate to the top of the hierarchy.	top	[edit security zones] user@host# **top** [edit] user@host#
Commit or Revert Changes		
Commit your configuration.	commit	[edit] user@host# **commit** commit complete
Roll back changes from the current session. Use the **rollback** command to revert all changes from the current configuration session. When you run the **rollback** command before exiting your session or committing changes, the software loads the most recently committed configuration onto the device. You must enter the **rollback** statement at the **edit** level in the hierarchy.	rollback	[edit] user@host# **rollback** load complete
Exit Configuration Mode		

Copyright © 2014, Juniper Networks, Inc.

Task	Command/Statement	Example
Commit the configuration and exit configuration mode.	commit and-quit	[edit] user@host# **commit and-quit** user@host>
Exit configuration mode without committing your configuration. You must navigate to the top of the hierarchy using the **up** or **top** commands before you can exit configuration mode.	exit	[edit] user@host# **exit** The configuration has been changed but not committed Exit with uncommitted changes? [yes,no] (yes)
Get Help		
Display a list of valid options for the current hierarchy level.	?	[edit] user@host# **edit security zones ?** Possible completions: <[Enter]> Execute this command > functional-zone Functional zone > security-zone Security zones | Pipe through a command [edit]

Related Documentation

- Understanding Junos OS CLI Configuration Mode on page 58
- Entering and Exiting the Junos OS CLI Configuration Mode on page 64
- Displaying the Current Junos OS Configuration on page 87

Checking the Status of a Device Running Junos OS

You can use **show** commands to check the status of the device and monitor the activities on the device.

To help you become familiar with **show** commands:

- Type **show ?** to display the list of **show** commands you can use to monitor the router:

```
root@>  show ?
Possible completions:
  accounting           Show accounting profiles and records
  aps                  Show Automatic Protection Switching information
  arp                  Show system Address Resolution Protocol table entries
  as-path              Show table of known autonomous system paths
  bfd                  Show Bidirectional Forwarding Detection information
  bgp                  Show Border Gateway Protocol information
  chassis              Show chassis information
  class-of-service     Show class-of-service (CoS) information
  cli                  Show command-line interface settings
  configuration        Show current configuration
  connections          Show circuit cross-connect connections
  dvmrp                Show Distance Vector Multicast Routing Protocol
info
  dynamic-tunnels      Show dynamic tunnel information information
  esis                 Show end system-to-intermediate system information
```


Copyright © 2014, Juniper Networks, Inc.

```
  firewall             Show firewall information
  helper               Show port-forwarding helper information
  host                 Show hostname information from domain name server
  igmp                 Show Internet Group Management Protocol information
  ike                  Show Internet Key Exchange information
  ilmi                 Show interim local management interface information
  interfaces           Show interface information
  ipsec                Show IP Security information
  ipv6                 Show IP version 6 information
  isis                 Show Intermediate System-to-Intermediate System info
  l2circuit            Show Layer 2 circuit information
  l2vpn                Show Layer 2 VPN information
  lacp                 Show Link Aggregation Control Protocol information
  ldp                  Show Label Distribution Protocol information
  link-management      Show link management information
  llc2                 Show LLC2 protocol related information
  log                  Show contents of log file
  mld                  Show multicast listener discovery information
  mpls                 Show Multiprotocol Label Switching information
  msdp                 Show Multicast Source Discovery Protocol information
  multicast            Show multicast information
  ntp                  Show Network Time Protocol information
  ospf                 Show Open Shortest Path First information
  ospf3                Show Open Shortest Path First version 3 information
  passive-monitoring   Show information about passive monitoring
  pfe                  Show Packet Forwarding Engine information
  pgm                  Show Pragmatic Generalized Multicast information
  pim                  Show Protocol Independent Multicast information
  policer              Show interface policer counters and information
  policy               Show policy information
  ppp                  Show PPP process information
  rip                  Show Routing Information Protocol information
  ripng                Show Routing Information Protocol for IPv6 info
  route                Show routing table information
  rsvp                 Show Resource Reservation Protocol information
  sap                  Show Session Announcement Protocol information
  security             Show security information
  services             Show services information
  snmp                 Show Simple Network Management Protocol information
  system               Show system information
  task                 Show routing protocol per-task information
  ted                  Show Traffic Engineering Database information
  version              Show software process revision levels
  vpls                 Show VPLS information
  vrrp                 Show Virtual Router Redundancy Protocol information
```

- Use the **show chassis routing-engine** command to view the Routing Engine status:

```
root@> show chassis routing-engine
Routing Engine status:
  Slot 0:
    Current state                  Master
    Election priority              Master (default)
    Temperature                 31 degrees C / 87 degrees F
    CPU temperature             32 degrees C / 89 degrees F
    DRAM                       768 MB
    Memory utilization          84 percent
    CPU utilization:
      User                       0 percent
      Background                 0 percent
      Kernel                     1 percent
      Interrupt                  0 percent
```

```
  Idle                          99 percent
Model                           RE-2.0
Serial ID                       b10000078c10d701
Start time                      2005-12-28 13:52:00 PST
Uptime                         12 days, 3 hours, 44 minutes, 19 seconds
Load averages:                  1 minute   5 minute  15 minute
                                    0.02       0.01       0.00
```

- Use the **show system storage** command to view available storage on the device:

```
root@> show system storage

Filesystem        Size       Used     Avail  Capacity   Mounted on
/dev/ad0s1a       865M       127M      669M       16%  /
devfs             1.0K       1.0K        0B      100%  /dev
devfs             1.0K       1.0K        0B      100%  /dev/
/dev/md0          30M        30M         0B      100%  /packages/mnt/jbase
/dev/md1          158M       158M        0B      100%
/packages/mnt/jkernel-9.3B1.5
/dev/md2          16M        16M         0B      100%
/packages/mnt/jpfe-M7i-9.3B1.5
/dev/md3          3.8M       3.8M        0B      100%
/packages/mnt/jdocs-9.3B1.5
/dev/md4          44M        44M         0B      100%
/packages/mnt/jroute-9.3B1.5
/dev/md5          12M        12M         0B      100%
/packages/mnt/jcrypto-9.3B1.5
/dev/md6          25M        25M         0B      100%
/packages/mnt/jpfe-common-9.3B1.5
/dev/md7          1.5G       196K      1.4G        0%  /tmp
/dev/md8          1.5G       910K      1.4G        0%  /mfs
/dev/ad0s1e       96M        38K        88M        0%  /config
procfs            4.0K       4.0K        0B      100%  /proc
/dev/ad1s1f       17G        2.6G       13G       17%  /var
```

Related Documentation

- Displaying the Junos OS CLI Command and Word History on page 56
- Managing Programs and Processes Using Junos OS Operational Mode Commands on page 165
- Viewing Files and Directories on a Device Running Junos OS on page 159

Configuring a Routing Protocol

This topic provides a sample configuration that describes how to configure an OSPF backbone area that has two SONET interfaces.

The final configuration looks like this:

```
[edit]
protocols {
  ospf {
    area 0.0.0.0 {
      interface so-0/0/0 {
        hello-interval 5;
        dead-interval 20;
      }
      interface so-0/0/1 {
```


Copyright © 2014, Juniper Networks, Inc.

```
            hello-interval 5;
            dead-interval 20;
        }
    }
  }
}
```

This topic contains the following examples of configuring a routing protocol:

- Shortcut on page 41
- Longer Configuration on page 41
- Making Changes to a Routing Protocol Configuration on page 43

Shortcut

You can create a shortcut for this entire configuration with the following two commands:

```
[edit]
user@host# set protocols ospf area 0.0.0.0 interface so-0/0/0 hello-interval 5
  dead-interval 20
[edit]
user@host# set protocols ospf area 0.0.0.0 interface so-0/0/1 hello-interval 5
  dead-interval 20
```

Longer Configuration

This section provides a longer example of creating the previous OSPF configuration. In the process, it illustrates how to use the different features of the CLI.

1. Enter configuration mode by issuing the **configure** top-level command:

   ```
   user@host> configure
   entering configuration mode
   [edit]
   user@host#
   ```

   Notice that the prompt has changed to a pound sign (**#**) to indicate configuration mode.

2. To create the above configuration, you start by editing the **protocols ospf** statements:

   ```
   [edit]
   user@host# edit protocols ospf
   [edit protocols ospf]
   user@host#
   ```

3. Now add the OSPF area:

   ```
   [edit protocols ospf]
   user@host# edit area 0.0.0.0
   [edit protocols ospf area 0.0.0.0]
   user@host#
   ```

4. Add the first interface:

   ```
   [edit protocols ospf area 0.0.0.0]
   user@host# edit interface so0
   [edit protocols ospf area 0.0.0.0 interface so-0/0/0]
   ```

Copyright © 2014, Juniper Networks, Inc.

```
user@host#
```

You now have four nested statements.

5. Set the hello and dead intervals.

```
[edit protocols ospf area 0.0.0.0 interface so-0/0/0]
user@host#set ?
user@host# set hello-interval 5
user@host# set dead-interval 20
user@host#
```

6. You can see what is configured at the current level with the **show** command:

```
[edit protocols ospf area 0.0.0.0 interface so-0/0/0]
user@host# show
hello-interval 5;
dead-interval 20;
[edit protocols ospf area 0.0.0.0 interface so-0/0/0]
user@host#
```

7. You are finished at this level, so back up a level and take a look at what you have so far:

```
[edit protocols ospf area 0.0.0.0 interface so-0/0/0]
user@host# up
[edit protocols ospf area 0.0.0.0]
user@host# show
interface so-0/0/0 {
  hello-interval 5;
  dead-interval 20;
}
[edit protocols ospf area 0.0.0.0]
user@host#
```

The **interface** statement appears because you have moved to the **area** statement.

8. Add the second interface:

```
[edit protocols ospf area 0.0.0.0]
user@host# edit interface so-0/0/1
[edit protocols ospf area 0.0.0.0 interface so-0/0/1]
user@host# set hello-interval 5
[edit protocols ospf area 0.0.0.0 interface so-0/0/1]
user@host# set dead-interval 20
[edit protocols ospf area 0.0.0.0 interface so-0/0/1]
user@host# up
[edit protocols ospf area 0.0.0.0]
user@host# show
interface so-0/0/0 {
  hello-interval 5;
  dead-interval 20;
}
interface so-0/0/1 {
  hello-interval 5;
  dead-interval 20;
}
[edit protocols ospf area 0.0.0.0]
user@host#
```


Copyright © 2014, Juniper Networks, Inc.

9. Back up to the top level and see what you have:

```
[edit protocols ospf area 0.0.0.0]
user@host# top
[edit]
user@host# show
protocols {
  ospf {
    area 0.0.0.0 {
      interface so-0/0/0 {
        hello-interval 5;
        dead-interval 20;
      }
      interface so-0/0/1 {
        hello-interval 5;
        dead-interval 20;
      }
    }
  }
}
[edit]
user@host#
```

   This configuration now contains the statements you want.

10. Before committing the configuration (and thereby activating it), verify that the configuration is correct:

```
[edit]
user@host# commit check
configuration check succeeds
[edit]
user@host#
```

11. Commit the configuration to activate it on the router:

```
[edit]
user@host# commit
commit complete
[edit]
user@host#
```

Making Changes to a Routing Protocol Configuration

Suppose you decide to use different dead and hello intervals on interface **so-0/0/1**. You can make changes to the configuration.

1. Go directly to the appropriate hierarchy level by typing the full hierarchy path to the statement you want to edit:

```
[edit]
user@host# edit protocols ospf area 0.0.0.0 interface so-0/0/1
[edit protocols ospf area 0.0.0.0 interface so-0/0/1]
user@host# show
hello-interval 5;
dead-interval 20;
[edit protocols ospf area 0.0.0.0 interface so-0/0/1]
```

Copyright © 2014, Juniper Networks, Inc.

```
user@host# set hello-interval 7
[edit protocols ospf area 0.0.0.0 interface so-0/0/1]
user@host# set dead-interval 28
[edit protocols ospf area 0.0.0.0 interface so-0/0/1]
user@host# top
[edit]
user@host# show
protocols {
  ospf {
    area 0.0.0.0 {
      interface so-0/0/0 {
        hello-interval 5;
        dead-interval 20;
      }
      interface so-0/0/1 {
        hello-interval 7;
        dead-interval 28;
      }
    }
  }
}
[edit]
user@host#
```

2. If you decide not to run OSPF on the first interface, delete the statement:

```
[edit]
user@host# edit protocols ospf area 0.0.0.0
[edit protocols ospf area 0.0.0.0]
user@host# delete interface so-0/0/0
[edit protocols ospf area 0.0.0.0]
user@host# top
[edit]
user@host# show
protocols {
  ospf {
    area 0.0.0.0 {
      interface so-0/0/1 {
        hello-interval 7;
        dead-interval 28;
      }
    }
  }
}
[edit]
user@host#
```

   Everything inside the statement you deleted was deleted with it. You can also eliminate the entire OSPF configuration by simply entering **delete protocols ospf** while at the top level.

3. If you decide to use the default values for the hello and dead intervals on your remaining interface but you want OSPF to run on that interface, delete the hello and dead interval timers:

```
[edit]
user@host# edit protocols ospf area 0.0.0.0 interface so-0/0/1
```


Copyright © 2014, Juniper Networks, Inc.

```
[edit protocols ospf area 0.0.0.0 interface so-0/0/1]
user@host# delete hello-interval
[edit protocols ospf area 0.0.0.0 interface so-0/0/1]
user@host# delete dead-interval
[edit protocols ospf area 0.0.0.0 interface so-0/0/1]
user@host# top
[edit]
user@host# show
protocols {
  ospf {
    area 0.0.0.0 {
      interface so-0/0/1;
    }
  }
}
[edit]
user@host#
```

You can set multiple statements at the same time as long as they are all part of the same hierarchy (the path of statements from the top inward, as well as one or more statements at the bottom of the hierarchy). This feature can reduce considerably the number of commands you must enter.

4. To go back to the original hello and dead interval timers on interface **so-0/0/1**, enter:

```
[edit]
user@host# edit protocols ospf area 0.0.0.0 interface so-0/0/1
[edit protocols ospf area 0.0.0.0 interface so-0/0/1]
user@host# set hello-interval 5 dead-interval 20
[edit protocols ospf area 0.0.0.0 interface so-0/0/1]
user@host# exit
[edit]
user@host# show
protocols {
  ospf {
    area 0.0.0.0 {
      interface so-0/0/1 {
        hello-interval 5;
        dead-interval 20;
      }
    }
  }
}
[edit]
user@host#
```

5. You also can re-create the other interface, as you had it before, with only a single entry:

```
[edit]
user@host# set protocols ospf area 0.0.0.0 interface so-0/0/1 hello-interval 5
  dead-interval 20
[edit]
user@host# show
protocols {
  ospf {
    area 0.0.0.0 {
```

Copyright © 2014, Juniper Networks, Inc.

```
            interface so-0/0/0 {
              hello-interval 5;
              dead-interval 20;
            }
            interface so-0/0/1 {
              hello-interval 5;
              dead-interval 20;
            }
          }
        }
      }
      [edit]
      user@host#
```

Related Documentation

- Getting Started with the Junos OS Command-Line Interface on page 31
- Displaying the Junos OS CLI Command and Word History on page 56
- Interface Naming Conventions Used in the Junos OS Operational Commands on page 158

Rolling Back Junos OS Configuration Changes

This topic shows how to use the **rollback** command to return to the most recently committed Junos OS configuration. The **rollback** command is useful if you make configuration changes and then decide not to keep the changes.

The following procedure shows how to configure an SNMP health monitor on a device running Junos OS and then return to the most recently committed configuration that does not include the health monitor. When configured, the SNMP health monitor provides the network management system (NMS) with predefined monitoring for file system usage, CPU usage, and memory usage on the device.

1. Enter configuration mode:

   ```
   user@host> configure
   entering configuration mode
   [edit]
   user@host#
   ```

2. Show the current configuration (if any) for SNMP:

   ```
   [edit]
   user@host# show snmp
   ```

   No **snmp** statements appear because SNMP has not been configured on the device.

3. Configure the health monitor:

   ```
   [edit]
   user@host# set snmp health-monitor
   ```

4. Show the new configuration:

   ```
   [edit]
   user@host# show snmp
   health-monitor;
   ```


Copyright © 2014, Juniper Networks, Inc.

The **health-monitor** statement indicates that SNMP health monitoring is configured on the device.

5. Enter the **rollback** configuration mode command to return to the most recently committed configuration:

   ```
   [edit]
   user@host# rollback
   load complete
   ```

6. Show the configuration again to make sure your change is no longer present:

   ```
   [edit]
   user@host# show snmp
   ```

   No **snmp** configuration statements appear. The health monitor is no longer configured.

7. Enter the **commit** command to activate the configuration to which you rolled back:

   ```
   [edit]
   user@host# commit
   ```

8. Exit configuration mode:

   ```
   [edit]
   user@host# exit
   Exiting configuration mode
   ```

You can also use the **rollback** command to return to earlier configurations.

Related Documentation

- Returning to the Most Recently Committed Junos OS Configuration on page 120

Copyright © 2014, Juniper Networks, Inc.

Copyright © 2014, Juniper Networks, Inc.

CHAPTER 3

Getting Online Help

- Getting Online Help from the Junos OS Command-Line Interface on page 49
- Junos OS CLI Online Help Features on page 52
- Examples: Using Command Completion in Configuration Mode on page 53
- Examples: Using the Junos OS CLI Command Completion on page 55
- Displaying the Junos OS CLI Command and Word History on page 56

Getting Online Help from the Junos OS Command-Line Interface

The Junos OS command-line interface (CLI) has a context-sensitive online help feature that enables you to access information about commands and statements from the Junos OS CLI. This topic contains the following sections:

- Getting Help About Commands on page 49
- Getting Help About a String in a Statement or Command on page 50
- Getting Help About Configuration Statements on page 51
- Getting Help About System Log Messages on page 51

Getting Help About Commands

Information about commands is provided at each level of the CLI command hierarchy. You can type a question mark to get help about commands:

- If you type the question mark at the command-line prompt, the CLI lists the available commands and options. For example, to view a list of top-level operational mode commands, type a question mark (?) at the command-line prompt.

```
user@host> ?
Possible completions:
clear       Clear information in the system
configure   Manipulate software configuration information
file        Perform file operations
help        Provide help information
mtrace      Trace mtrace packets from source to receiver.
monitor     Real-time debugging
ping        Ping a remote target
quit        Exit the management session
request     Make system-level requests
restart     Restart a software process
set         Set CLI properties, date, time, craft display text
```

Copyright © 2014, Juniper Networks, Inc.

```
show        Show information about the system
ssh         Open a secure shell to another host
start       Start a software process
telnet      Telnet to another host
test        Diagnostic debugging commands
traceroute  Trace the route to a remote host
user@host>
```

- If you type the question mark after entering the complete name of a command or command option, the CLI lists the available commands and options and then redisplays the command names and options that you typed.

```
user@host> clear ?
Possible completions:
arp        Clear address-resolution information
bgp          Clear BGP information
chassis      Clear chassis information
firewall     Clear firewall counters
igmp         Clear IGMP information
interfaces   Clear interface information
ilmi         Clear ILMI statistics information
isis         Clear IS-IS information
ldp          Clear LDP information
log          Clear contents of a log file
mpls         Clear MPLS information
msdp         Clear MSDP information
multicast    Clear Multicast information
ospf         Clear OSPF information
pim          Clear PIM information
rip          Clear RIP information
route        Clear routing table information
rsvp         Clear RSVP information
snmp         Clear SNMP information
system       Clear system status
vrrp         Clear VRRP statistics information
user@host> clear
```

- If you type the question mark in the middle of a command name, the CLI lists possible command completions that match the letters you have entered so far. It then redisplays the letters that you typed. For example, to list all operational mode commands that start with the letter *c*, type the following:

```
user@host> c?
Possible completions:
clear      Clear information in the system
configure  Manipulate software configuration information
user@host> c
```

- For introductory information on using the question mark or the help command, you can also type **help** and press Enter:

```
user@host> help
```

Getting Help About a String in a Statement or Command

You can use the **help** command to display help about a text string contained in a statement or command name:

help apropos ***string***

Copyright © 2014, Juniper Networks, Inc.

string is a text string about which you want to get help. This string is used to match statement or command names as well as to match the help strings that are displayed for the statements or commands.

If the string contains spaces, enclose it in quotation marks (" "). You can also specify a regular expression for the string, using standard UNIX-style regular expression syntax.

For statements or commands which need input data type as STRING, the supported characters set are as follows:

- Any printable ASCII characters
- For characters with space, it should be enclosed in double-quotes
- To have double-quote as the input, it should be escaped with '\'

NOTE: **No escape characters are supported in a string other than to escape from double quotes.**

Range of supported characters for attributes is 0 through 65499 characters.

Range of supported characters for string type identifiers is 1 through 255 characters.

In configuration mode, this command displays statement names and help text that match the string specified. In operational mode, this command displays command names and help text that match the string specified.

Getting Help About Configuration Statements

You can display help based on text contained in a statement name using the **help topic** and **help reference** commands:

help topic ***word***
help reference ***statement-name***

The **help topic** command displays usage guidelines for the statement based on information that appears in the Junos OS configuration guides. The **help reference** command displays summary information about the statement based on the summary descriptions that appear in the Junos OS configuration guides.

Getting Help About System Log Messages

You can display help based on a system log tag using the **help syslog** command:

help syslog ***syslog-tag***

The **help syslog** command displays the contents of a system log message.

Related Documentation

- Junos OS CLI Online Help Features on page 52
- Getting Started with the Junos OS Command-Line Interface on page 31

Copyright © 2014, Juniper Networks, Inc.

Junos OS CLI Online Help Features

The Junos OS CLI online help provides the following features for ease of use and error prevention:

- Help for Omitted Statements on page 52
- Using CLI Command Completion on page 52
- Using Command Completion in Configuration Mode on page 53
- Displaying Tips About CLI Commands on page 53

Help for Omitted Statements

If you have omitted a required statement at a particular hierarchy level, when you attempt to move from that hierarchy level or when you issue the **show** command in configuration mode, a message indicates which statement is missing. For example:

```
[edit protocols pim interface so-0/0/0]
user@host# top
Warning: missing mandatory statement: 'mode'
[edit]
user@host# show
protocols {
   pim {
      interface so-0/0/0 {
         priority 4;
         version 2;
         # Warning: missing mandatory statement(s): 'mode'
      }
   }
}
```

Using CLI Command Completion

The Junos OS CLI provides you a command completion option that enables Junos OS to recognize commands and options based on the initial few letters you typed. That is, you do not always have to remember or type the full command or option name for the CLI to recognize it.

- To display all possible command or option completions, type the partial command followed immediately by a question mark.
- To complete a command or option that you have partially typed, press Tab or the Spacebar. If the partially typed letters begin a string that uniquely identifies a command, the complete command name appears. Otherwise, a prompt indicates that you have entered an ambiguous command, and the possible completions are displayed.

Command completion also applies to other strings, such as filenames, interface names, and usernames. To display all possible values, type a partial string followed immediately by a question mark. To complete a string, press Tab.

Copyright © 2014, Juniper Networks, Inc.

Using Command Completion in Configuration Mode

The CLI command completion functions also apply to the commands in configuration mode and to configuration statements. Specifically, to display all possible commands or statements, type the partial string followed immediately by a question mark. To complete a command or statement that you have partially typed, press Tab or the Spacebar.

Command completion also applies to identifiers, with one slight difference. To display all possible identifiers, type a partial string followed immediately by a question mark. To complete an identifier, you must press Tab. This scheme allows you to enter identifiers with similar names; then press the Spacebar when you are done typing the identifier name.

Displaying Tips About CLI Commands

To get tips about CLI commands, issue the **help tip cli** command. Each time you enter the command, a new tip appears. For example:

```
user@host> help tip cli
Junos tip:
Use 'request system software validate' to validate the incoming software
against the current configuration without impacting the running system.
user@host> help tip cli
Junos tip:
Use 'commit and-quit' to exit configuration mode after the commit has
succeeded. If the commit fails, you are left in configuration mode.
```

You can also enter **help tip cli *number*** to associate a tip with a number. This enables you to recall the tip at a later time. For example:

```
user@host> help tip cli 10
JUNOS tip:
Use '#' in the beginning of a line in command scripts to cause the
rest of the line to be ignored.

user@host> help tip cli
JUNOS tip:
Use the 'apply-groups' statement at any level of the configuration
hierarchy to inherit configuration statements from a configuration group.

user@host>
```

Related Documentation

- Getting Started with the Junos OS Command-Line Interface on page 31
- Examples: Using the Junos OS CLI Command Completion on page 55

Examples: Using Command Completion in Configuration Mode

List the configuration mode commands:

```
[edit]
user@host# ?
  <[Enter]>           Execute this command
  activate            Remove the inactive tag from a statement
```

Copyright © 2014, Juniper Networks, Inc.

```
  annotate             Annotate the statement with a comment
  commit               Commit current set of changes
  copy                 Copy a statement
  deactivate           Add the inactive tag to a statement
  delete               Delete a data element
  edit                 Edit a sub-element
  exit                 Exit from this level
  extension            Extension operations
  help                 Provide help information
  insert               Insert a new ordered data element
  load                 Load configuration from ASCII file
  quit                 Quit from this level
  rename               Rename a statement
  replace              Replace character string in configuration
  rollback             Roll back to previous committed configuration
  run                  Run an operational-mode command
  save                 Save configuration to ASCII file
  set                  Set a parameter
  show                 Show a parameter
  status               Show users currently editing configuration
  top                  Exit to top level of configuration
  up                   Exit one level of configuration
  wildcard             Wildcard operations
[edit]user@host#
```

List all the statements available at a particular hierarchy level:

```
[edit]
user@host# edit ?
Possible completions:
> accounting-options  Accounting data configuration
> chassis             Chassis configuration
> class-of-service    Class-of-service configuration
> firewall            Define a firewall configuration
> forwarding-options  Configure options to control packet sampling
> groups              Configuration groups
> interfaces          Interface configuration
> policy-options      Routing policy option configuration
> protocols           Routing protocol configuration
> routing-instances   Routing instance configuration
> routing-options     Protocol-independent routing option configuration
> snmp                Simple Network Management Protocol
> system              System parameters

user@host# edit protocols ?
Possible completions:
<[Enter]>          Execute this command
> bgp              BGP options
> connections      Circuit cross-connect configuration
> dvmrp            DVMRP options
> igmp             IGMP options
> isis             IS-IS options
> ldp              LDP options
> mpls             Multiprotocol Label Switching options
> msdp             MSDP options
> ospf             OSPF configuration
> pim              PIM options
> rip              RIP options
> router-discovery ICMP router discovery options
> rsvp             RSVP options
> sapSession       Advertisement Protocol options
```


Copyright © 2014, Juniper Networks, Inc.

```
> vrrp              VRRP options
|                   Pipe through a command

[edit]
user@host# edit protocols
```

List all commands that start with a particular letter or string:

```
user@host# edit routing-options a?
Possible completions:
> aggregate           Coalesced routes
> autonomous-system   Autonomous system number

[edit]
user@host# edit routing-options a
```

List all configured Asynchronous Transfer Mode (ATM) interfaces:

```
[edit]
user@host# edit interfaces at?
<interface_name>     Interface name
  at-0/2/0             Interface name
  at-0/2/1             Interface name
[edit]
user@host# edit interfaces at
```

Display a list of all configured policy statements:

```
[edit]
user@host# show policy-options policy-statement ?

Possible completions:
  <policy_name>        Name to identify a policy filter
[edit]
user@host# show policy-options policy-statement
```

Related Documentation

- Examples: Using the Junos OS CLI Command Completion on page 55
- Displaying the Junos OS CLI Command and Word History on page 56

Examples: Using the Junos OS CLI Command Completion

The following examples show how you can use the command completion feature in Junos OS. Issue the **show interfaces** command:

```
user@host> sh<Space>ow i<Space>
'i' is ambiguous.
Possible completions:
igmp                  Show information about IGMP
interface             Show interface information
isis                  Show information about IS-IS

user@host> show in<Space>terfaces
Physical interface: at-0/1/0, Enabled, Physical link is Up
Interface index: 11, SNMP ifIndex: 65
Link-level type: ATM-PVC, MTU: 4482, Clocking: Internal, SONET mode
Speed: OC12, Loopback: None, Payload scrambler: Enabled
Device flags: Present Running
Link flags: 0x01
...
```

Copyright © 2014, Juniper Networks, Inc.

```
user@host>
```

Display a list of all log files whose names start with the string “messages,” and then display the contents of one of the files:

```
user@myhost> show log mes?
Possible completions:
  <filename>Log file to display
messagesSize: 1417052, Last changed: Mar 3 00:33
messages.0.gzSize:  145575, Last changed: Mar 3 00:00
messages.1.gzSize:  134253, Last changed: Mar 2 23:00
messages.10.gzSize: 137022, Last changed: Mar 2 14:00
messages.2.grSize:  137112, Last changed: Mar 2 22:00
messages.3.gzSize:  121633, Last changed: Mar 2 21:00
messages.4.gzSize:  135715, Last changed: Mar 2 20:00
messages.5.gzSize:  137504, Last changed: Mar 2 19:00
messages.6.gzSize:  134591, Last changed: Mar 2 18:00
messages.7.gzSize:  132670, Last changed: Mar 2 17:00
messages.8.gzSize:  136596, Last changed: Mar 2 16:00
messages.9.gzSize:  136210, Last changed: Mar 2 15:00
```

```
user@myhost> show log mes<Tab>sages.4<Tab>.gz<Enter>
Jan 15 21:00:00 myhost newsyslog[1381]: logfile turned over
...
```

Related Documentation

- Displaying the Junos OS CLI Command and Word History on page 56

Displaying the Junos OS CLI Command and Word History

To display a list of recent commands that you issued, use the **show cli history** command:

```
user@host> show cli history 3
01:01:44 -- show bgp next-hop-database
01:01:51 -- show cli history
01:02:51 -- show cli history 3
```

You can press Esc+. (period) or Alt+. (period) to insert the last word of the previous command. Repeat Esc+. or Alt+. to scroll backwards through the list of recently entered words. For example:

```
user@host> show interfaces terse fe-0/0/0
Interface       Admin    Link    Proto     Local    Remote
fe-0/0/0        up       up
fe-0/0/0.0      up       up      inet     192.168.220.1/30
```

```
user@host> <Esc>
user@host> fe-0/0/0
```

If you scroll completely to the beginning of the list, pressing Esc+. or Alt+. again restarts scrolling from the last word entered.

Related Documentation

- Junos OS CLI Online Help Features on page 52

Copyright © 2014, Juniper Networks, Inc.

CHAPTER 4

Using Configuration Statements to Configure a Device

- Understanding Junos OS CLI Configuration Mode on page 58
- Entering and Exiting the Junos OS CLI Configuration Mode on page 64
- Forms of the configure Command on page 66
- Using the configure exclusive Command on page 67
- Example: Using the configure Command on page 68
- Modifying the Junos OS Configuration on page 69
- Adding Junos OS Configuration Statements and Identifiers on page 69
- Deleting a Statement from a Junos OS Configuration on page 71
- Example: Deleting a Statement from the Junos OS Configuration on page 72
- Copying a Junos OS Statement in the Configuration on page 73
- Example: Copying a Statement in the Junos OS Configuration on page 74
- Issuing Relative Junos OS Configuration Mode Commands on page 74
- Renaming an Identifier in a Junos OS Configuration on page 75
- Example: Renaming an Identifier in a Junos OS Configuration on page 75
- Inserting a New Identifier in a Junos OS Configuration on page 75
- Example: Inserting a New Identifier in a Junos OS Configuration on page 76
- Example: Using the Wildcard Command with the Range Option on page 78
- Deactivating and Reactivating Statements and Identifiers in a Junos OS Configuration on page 82
- Examples: Deactivating and Reactivating Statements and Identifiers in a Junos OS Configuration on page 83
- Adding Comments in a Junos OS Configuration on page 84
- Example: Including Comments in a Junos OS Configuration on page 85
- Updating the configure private Configuration on page 86
- Displaying the Current Junos OS Configuration on page 87
- Example: Displaying the Current Junos OS Configuration on page 88
- Displaying Additional Information About the Junos OS Configuration on page 89

Copyright © 2014, Juniper Networks, Inc.

- Displaying set Commands from the Junos OS Configuration on page 92
- Displaying Users Currently Editing the Junos OS Configuration on page 94
- Verifying a Junos OS Configuration on page 95

Understanding Junos OS CLI Configuration Mode

You can configure all properties of Junos OS, including interfaces, general routing information, routing protocols, and user access, as well as several system hardware properties.

As described in "Understanding the Junos OS CLI Modes, Commands, and Statement Hierarchies" on page 25, a router configuration is stored as a hierarchy of statements. In configuration mode, you create the specific hierarchy of configuration statements that you want to use. When you have finished entering the configuration statements, you commit them, which activates the configuration on the router.

You can create the hierarchy interactively or you can create an ASCII text file that is loaded onto the router or switch and then committed.

This topic covers:

- Configuration Mode Commands on page 59
- Configuration Statements and Identifiers on page 60
- Configuration Statement Hierarchy on page 62

Copyright © 2014, Juniper Networks, Inc.

Configuration Mode Commands

Table 4 on page 59 summarizes each CLI configuration mode command. The commands are organized alphabetically.

Table 4: Summary of Configuration Mode Commands

Command	Description
activate	Remove the **inactive:** tag from a statement, effectively reading the statement or identifier to the configuration. Statements or identifiers that have been activated take effect when you next issue the **commit** command.
annotate	Add comments to a configuration. You can add comments only at the current hierarchy level.
commit	Commit the set of changes to the database and cause the changes to take operational effect.
copy	Make a copy of an existing statement in the configuration.
deactivate	Add the **inactive:** tag to a statement, effectively commenting out the statement or identifier from the configuration. Statements or identifiers marked as inactive do not take effect when you issue the **commit** command.
delete	Delete a statement or identifier. All subordinate statements and identifiers contained within the specified statement path are deleted with it.
edit	Move inside the specified statement hierarchy. If the statement does not exist, it is created.
exit	Exit the current level of the statement hierarchy, returning to the level prior to the last edit command, or exit from configuration mode. The **quit** and **exit** commands are synonyms.
extension	Manage configurations that are contributed by SDK application packages. Either display or delete user-defined configuration contributed by the named SDK application package. A configuration defined in any native Junos OS package is never deleted by the extension command.
help	Display help about available configuration statements.
insert	Insert an identifier into an existing hierarchy.
load	Load a configuration from an ASCII configuration file or from terminal input. Your current location in the configuration hierarchy is ignored when the load operation occurs.

Copyright © 2014, Juniper Networks, Inc.

Table 4: Summary of Configuration Mode Commands *(continued)*

Command	Description
quit	Exit the current level of the statement hierarchy, returning to the level prior to the last edit command, or exit from configuration mode. The **quit** and **exit** commands are synonyms.
rename	Rename an existing configuration statement or identifier.
replace	Replace identifiers or values in a configuration.
rollback	Return to a previously committed configuration. The software saves the last 10 committed configurations, including the rollback number, date, time, and name of the user who issued the **commit configuration** command.
run	Run a top-level CLI command without exiting from configuration mode.
save	Save the configuration to an ASCII file. The contents of the current level of the statement hierarchy (and below) are saved, along with the statement hierarchy containing it. This allows a section of the configuration to be saved, while fully specifying the statement hierarchy.
set	Create a statement hierarchy and set identifier values. This is similar to **edit** except that your current level in the hierarchy does not change.
show	Display the current configuration.
status	Display the users currently editing the configuration.
top	Return to the top level of configuration command mode, which is indicated by the **[edit]** banner.
up	Move up one level in the statement hierarchy.
update	Update a private database.
wildcard	Delete a statement or identifier. All subordinate statements and identifiers contained within the specified statement path are deleted with it. You can use regular expressions to specify a pattern. Based on this pattern, you search for items that contain these patterns and delete them.

Configuration Statements and Identifiers

You can configure router or switch properties by including the corresponding statements in the configuration. Typically, a statement consists of a keyword, which is fixed text, and, optionally, an identifier. An identifier is an identifying name that you can define, such as

Copyright © 2014, Juniper Networks, Inc.

the name of an interface or a username, which enables you and the CLI to differentiate among a collection of statements.

Table 5 on page 61 describes top-level CLI configuration mode statements.

NOTE: The QFX3500 switch does not support the IS-IS, OSPF, BGP, LDP, MPLS, and RSVP protocols.

Table 5: Configuration Mode Top-Level Statements

Statement	Description
access	Configure the Challenge Handshake Authentication Protocol (CHAP). For information about the statements in this hierarchy, see the *Junos OS Administration Library for Routing Devices*.
accounting-options	Configure accounting statistics data collection for interfaces and firewall filters. For information about the statements in this hierarchy, see the *Network Management Administration Guide for Routing Devices*.
chassis	Configure properties of the router chassis, including conditions that activate alarms and SONET/SDH framing and concatenation properties. For information about the statements in this hierarchy, see the *Junos OS Administration Library for Routing Devices*.
class-of-service	Configure class-of-service parameters. For information about the statements in this hierarchy, see the *Class of Service Feature Guide for Routing Devices*.
firewall	Define filters that select packets based on their contents. For information about the statements in this hierarchy, see the *Routing Policies, Firewall Filters, and Traffic Policers Feature Guide for Routing Devices*.
forwarding-options	Define forwarding options, including traffic sampling options. For information about the statements in this hierarchy, see the *Junos OS Network Interfaces Library for Routing Devices*.
groups	Configure configuration groups. For information about statements in this hierarchy, see the *Junos OS Administration Library for Routing Devices*.
interfaces	Configure interface information, such as encapsulation, interfaces, virtual channel identifiers (VCIs), and data-link connection identifiers (DLCIs). For information about the statements in this hierarchy, see the *Junos OS Network Interfaces Library for Routing Devices*.
policy-options	Define routing policies, which allow you to filter and set properties in incoming and outgoing routes. For information about the statements in this hierarchy, see the *Routing Policies, Firewall Filters, and Traffic Policers Feature Guide for Routing Devices*.
protocols	Configure routing protocols, including BGP, IS-IS, LDP, MPLS, OSPF, RIP, and RSVP. For information about the statements in this hierarchy, see the chapters that discuss how to configure the individual routing protocols in the *Junos OS Routing Protocols Library for Routing Devices* and the *Junos OS MPLS Applications Library for Routing Devices*.

Copyright © 2014, Juniper Networks, Inc.

Table 5: Configuration Mode Top-Level Statements *(continued)*

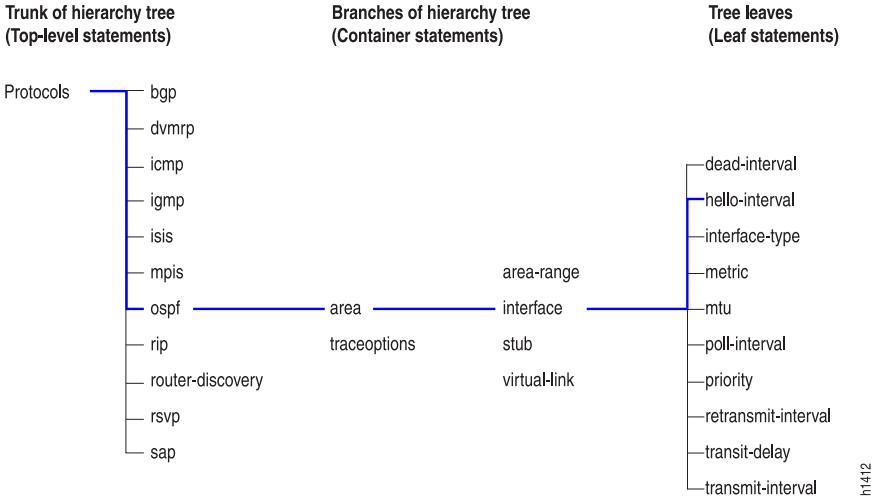
Statement	Description
routing-instances	Configure multiple routing instances. For information about the statements in this hierarchy, see the *Junos OS Routing Protocols Library for Routing Devices*.
routing-options	Configure protocol-independent routing options, such as static routes, autonomous system numbers, confederation members, and global tracing (debugging) operations to log. For information about the statements in this hierarchy, see the *Junos OS Routing Protocols Library for Routing Devices*.
security	Configure IP Security (IPsec) services. For information about the statements in this hierarchy see the *Junos OS Administration Library for Routing Devices*.
snmp	Configure SNMP community strings, interfaces, traps, and notifications. For information about the statements in this hierarchy, see the *Network Management Administration Guide for Routing Devices*.
system	Configure systemwide properties, including the hostname, domain name, Domain Name System (DNS) server, user logins and permissions, mappings between hostnames and addresses, and software processes. For information about the statements in this hierarchy, see the *Junos OS Administration Library for Routing Devices*.

For specific information on configuration statements, see the Junos OS configuration guides.

Configuration Statement Hierarchy

The Junos OS configuration consists of a hierarchy of *statements*. There are two types of statements: *container statements*, which are statements that contain other statements, and *leaf statements*, which do not contain other statements (see Figure 4 on page 62). All of the container and leaf statements together form the *configuration hierarchy*.

Figure 4: Configuration Mode Hierarchy of Statements



Copyright © 2014, Juniper Networks, Inc.

Each statement at the top level of the configuration hierarchy resides at the trunk (or root level) of a hierarchy tree. The top-level statements are container statements, containing other statements that form the tree branches. The leaf statements are the leaves of the hierarchy tree. An individual hierarchy of statements, which starts at the trunk of the hierarchy tree, is called a *statement path*. Figure 4 on page 62 illustrates the hierarchy tree, showing a statement path for the portion of the protocol configuration hierarchy that configures the hello interval on an interface in an OSPF area.

The **protocols** statement is a top-level statement at the trunk of the configuration tree. The **ospf**, **area**, and **interface** statements are all subordinate container statements of a higher statement (they are branches of the hierarchy tree); and the **hello-interval** statement is a leaf on the tree which in this case contains a data value: the length of the hello interval, in seconds.

The CLI represents the statement path shown in Figure 4 on page 62 as **[edit protocols ospf area *area-number* interface *interface-name*]** and displays the configuration as follows:

```
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0 {
                hello-interval 5;
            }
            interface so-0/0/1 {
                hello-interval 5;
            }
        }
    }
}
```

The CLI indents each level in the hierarchy to indicate each statement's relative position in the hierarchy and generally sets off each level with braces, using an open brace at the beginning of each hierarchy level and a closing brace at the end. If the statement at a hierarchy level is empty, the braces are not printed.

Each leaf statement ends with a semicolon. If the hierarchy does not extend as far as a leaf statement, the last statement in the hierarchy ends with a semicolon.

The configuration hierarchy can also contain "oneliners" at the last level in the hierarchy. Oneliners remove one level of braces in the syntax and display the container statement, its identifiers, the child or leaf statement and its attributes all on one line. For example, in the following sample configuration hierarchy, the line **level 1 metric *10*** is a oneliner because the **level** container statement with identifier ***1***, its child statement **metric**, and its corresponding attribute ***10*** all appear on a single line in the hierarchy:

```
[edit protocols]
    isis {
        interface ge-0/0/0.0 {
            level 1 metric 10;
        }
    }
}
```

Copyright © 2014, Juniper Networks, Inc.

Likewise, in the following example, **dynamic-profile** *dynamic-profile-name* **aggregate-clients;** is a oneliner because the **dynamic-profile** statement, its identifier ***dynamic-profile-name***, and leaf statement **aggregate-clients** all appear on one line when you run the **show** command in the configuration mode:

```
[edit forwarding-options]
user@host# show
dhcp-relay {
  dynamic-profile dynamic-profile-name aggregate-clients;
}
```

Related Documentation

- Entering and Exiting the Junos OS CLI Configuration Mode on page 64

Entering and Exiting the Junos OS CLI Configuration Mode

You configure Junos OS by entering configuration mode and creating a hierarchy of configuration mode statements.

- To enter configuration mode, use the **configure** command.

  When you enter configuration mode, the following configuration mode commands are available:

```
user@host>configure
entering configuration mode

[edit]
user@host#?
possible completions:
  <[Enter]>            Execute this command
  activate             Remove the inactive tag from a statement
  annotate             Annotate the statement with a comment
  commit               Commit current set of changes
  copy                 Copy a statement
  deactivate           Add the inactive tag to a statement
  delete               Delete a data element
  edit                 Edit a sub-element
  exit                 Exit from this level
  help                 Provide help information
  insert               Insert a new ordered data element
  load                 Load configuration from ASCII file
  quit                 Quit from this level
  rename               Rename a statement
  replace              Replace character string in configuration
  rollback             Roll back to previous committed configuration
  run                  Run an operational-mode command
  save                 Save configuration to ASCII file
  set                  Set a parameter
  show                 Show a parameter
  status               Show users currently editing configuration
  top                  Exit to top level of configuration
  up                   Exit one level of configuration
  wildcard             Wildcard operations
[edit]
user@host>
```


Copyright © 2014, Juniper Networks, Inc.

Users must have configure permission to view and use the **configure** command. When in configuration mode, a user can view and modify only those statements for which they have access privileges set. For more information, see the *Junos OS Administration Library for Routing Devices*.

- If you enter configuration mode and another user is also in configuration mode, a message shows the user's name and what part of the configuration the user is viewing or editing:

```
user@host> configure
Entering configuration mode
Users currently editing the configuration:
  root terminal d0 (pid 4137) on since 2008-04-09 23:03:07 PDT, idle 7w6d 08:22

  [edit]
The configuration has been changed but not committed

[edit]
user@host#
```

  Up to 32 users can be in configuration mode simultaneously, and they all can make changes to the configuration at the same time.

- To exit configuration mode, use the **exit configuration-mode** configuration mode command from any level, or use the **exit** command from the top level. For example:

```
[edit protocols ospf area 0.0.0.0 interface so-0/0/0]
user@host# exit configuration-mode
exiting configuration mode
user@host>
```

```
[edit]
user@host# exit
exiting configuration mode
user@host>
```

  If you try to exit from configuration mode using the **exit** command and the configuration contains changes that have not been committed, you see a message and prompt:

```
[edit]
user@host# exit
The configuration has been changed but not committed
Exit with uncommitted changes? [yes,no] (yes) <Enter>
Exiting configuration mode
user@host>
```

- To exit with uncommitted changes without having to respond to a prompt, use the **exit configuration-mode** command. This command is useful when you are using scripts to perform remote configuration.

```
[edit]
user@host# exit configuration-mode
The configuration has been changed but not committed
Exiting configuration mode
user@host>
```

Related Documentation

- Understanding Junos OS CLI Configuration Mode on page 58

Copyright © 2014, Juniper Networks, Inc.

- Modifying the Junos OS Configuration on page 69
- Commit Operation When Multiple Users Configure the Software on page 101
- Displaying the Current Junos OS Configuration on page 87
- Displaying set Commands from the Junos OS Configuration on page 92
- Issuing Relative Junos OS Configuration Mode Commands on page 74
- Using the configure exclusive Command on page 67
- Updating the configure private Configuration on page 86
- Switching Between Junos OS CLI Operational and Configuration Modes on page 33

Forms of the configure Command

The Junos OS supports three forms of the **configure** command: **configure**, **configure private**, and **configure exclusive**. These forms control how users edit and commit configurations and can be useful when multiple users configure the software. See Table 6 on page 66.

Table 6: Forms of the configure Command

Command	Edit Access	Commit Access
configure	• No one can lock the configuration. All users can make configuration changes. When you enter configuration mode, the CLI displays the following information: ◦ A list of other users editing the configuration. ◦ Hierarchy levels the users are viewing or editing. ◦ Whether the configuration has been changed, but not committed. ◦ When multiple users enter conflicting configurations, the most recent change to be entered takes precedence.	• No one can lock the configuration. All users can commit all changes to the configuration. • If you and another user make changes and the other user commits changes, your changes are committed as well.

Copyright © 2014, Juniper Networks, Inc.

Table 6: Forms of the configure Command *(continued)*

Command	Edit Access	Commit Access
configure exclusive	• One user locks the configuration and makes changes without interference from other users. • Other users can enter and exit configuration mode, but they cannot commit the configuration. • If you enter configuration mode while another user has locked the configuration (with the **configure exclusive** command), the CLI displays the user and the hierarchy level the user is viewing or editing. • If you enter configuration mode while another user has locked the configuration, you can forcibly log out that user with the **request system logout** operational mode command. For details, see the CLI Explorer.	
configure private	• Multiple users can edit the configuration at the same time. • Each user has a private candidate configuration to edit independently of other users. • When multiple users enter conflicting configurations, the first commit operation takes precedence over subsequent commit operations.	• When you commit the configuration, the router verifies that the operational (running) configuration has not been modified by another user before accepting your private candidate configuration as the new operational configuration. • If the configuration has been modified by another user, you can merge the modifications into your private candidate configuration and attempt to commit again.

Related Documentation

- Committing a Junos OS Configuration on page 98
- Example: Using the configure Command on page 68
- Displaying Users Currently Editing the Junos OS Configuration on page 94
- Using the configure exclusive Command on page 67
- Updating the configure private Configuration on page 86
- Displaying set Commands from the Junos OS Configuration on page 92

Using the configure exclusive Command

If you enter configuration mode with the **configure exclusive** command, you lock the candidate *global* configuration (also known as the *shared configuration* or *shared configuration database*) for as long as you remain in configuration mode, allowing you to make changes without interference from other users. Other users can enter and exit configuration mode, but they cannot commit the configuration.

If another user has locked the configuration, and you need to forcibly log the person out, enter the operational mode command **request system logout pid** ***pid_number***.

Copyright © 2014, Juniper Networks, Inc.

If you enter configuration mode and another user is also in configuration mode and has locked the configuration, a message identifies the user and the portion of the configuration that the user is viewing or editing:

```
user@host> configure
Entering configuration mode
Users currently editing the configuration:
root terminal p3 (pid 1088) on since 2000-10-30 19:47:58 EDT, idle 00:00:44
exclusive [edit interfaces so-3/0/0 unit 0 family inet]
```

In configure exclusive mode, any uncommitted changes are discarded when you exit:

```
user@host> configure exclusive
warning: uncommitted changes will be discarded on exit
Entering configuration mode
[edit]
user@host# set system host-name cool
[edit]
user@host# quit
The configuration has been changed but not committed
warning: Auto rollback on exiting 'configure exclusive'
Discard uncommitted changes? [yes,no] (yes)
warning: discarding uncommitted changes
load complete
Exiting configuration mode
```

When you use the **yes** option to exit configure exclusive mode, Junos OS discards your uncommitted changes and rolls backs your configuration. The **no** option allows you to continue editing or to commit your changes in configure exclusive mode.

When a user exits from configure exclusive mode while another user is in configure private mode, Junos OS will roll back any uncommitted changes.

Related Documentation

- Adding Junos OS Configuration Statements and Identifiers on page 69
- Forms of the configure Command on page 66

Example: Using the configure Command

If, when you enter configuration mode, another user is also in configuration mode, a message shows who the user is and what part of the configuration that user is viewing or editing:

```
user@host> configure
Entering configuration mode
Current configuration users:
root terminal p3 (pid 1088) on since 1999-05-13 01:03:27 EDT
[edit interfaces so-3/0/0 unit 0 family inet]
The configuration has been changed but not committed
[edit]
user@host#
```

If, when you enter configuration mode, the configuration contains changes that have not been committed, a message appears:

Copyright © 2014, Juniper Networks, Inc.

```
user@host> configure
Entering configuration mode
The configuration has been changed but not committed
[edit]
user@host#
```

Related Documentation

- Forms of the configure Command on page 66

Modifying the Junos OS Configuration

To configure a device running Junos OS or to modify an existing Junos OS configuration, you add statements to the configuration. For each statement hierarchy, you create the hierarchy starting with a statement at the top level and continuing with statements that move progressively lower in the hierarchy.

To modify the hierarchy, you use two configuration mode commands:

- **edit**—Moves to a particular hierarchy level. If that hierarchy level does not exist, the **edit** command creates it. The **edit** command has the following syntax:

  edit <***statement-path***>

- **set**—Creates a configuration statement and sets identifier values. After you issue a **set** command, you remain at the same level in the hierarchy. The **set** command has the following syntax:

  set <***statement-path***> ***statement*** <***identifier***>

  statement-path is the hierarchy to the configuration statement and the statement itself. If you have already moved to the statement's hierarchy level, you can omit the statement path. ***statement*** is the configuration statement itself. ***identifier*** is a string that identifies an instance of a statement.

You cannot use the **edit** command to change the value of identifiers. You must use the **set** command.

Related Documentation

- Displaying the Current Junos OS Configuration on page 87
- Adding Junos OS Configuration Statements and Identifiers on page 69
- Using the configure exclusive Command on page 67
- Updating the configure private Configuration on page 86
- Issuing Relative Junos OS Configuration Mode Commands on page 74

Adding Junos OS Configuration Statements and Identifiers

All properties of a device running Junos OS are configured by including *statements* in the configuration. A statement consists of a keyword, which is fixed text, and, optionally, an *identifier*. An identifier is an identifying name which you define, such as the name of an interface or a username, and which allows you and the CLI to discriminate among a collection of statements.

Copyright © 2014, Juniper Networks, Inc.

For example, the following list shows the statements available at the top level of configuration mode:

```
user@host# set?
Possible completions:
> accounting-options   Accounting data configuration
+ apply-groups         Groups from which to inherit configuration data
> chassis              Chassis configuration
> class-of-service     Class-of-service configuration
> firewall             Define a firewall configuration
> forwarding-options   Configure options to control packet sampling
> groups               Configuration groups
> interfaces           Interface configuration
> policy-options       Routing policy option configuration
> protocols            Routing protocol configuration
> routing-instances    Routing instance configuration
> routing-options      Protocol-independent routing option configuration
> snmp                 Simple Network Management Protocol
> system               System parameters
```

An angle bracket (>) before the statement name indicates that it is a container statement and that you can define other statements at levels below it. If there is no angle bracket (>) before the statement name, the statement is a leaf statement; you cannot define other statements at hierarchy levels below it.

A plus sign (+) before the statement name indicates that it can contain a set of values. To specify a set, include the values in brackets. For example:

```
[edit]
user@host# set policy-options community my-as1-transit members [65535:10 65535:11]
```

In some statements, you can include an identifier. For some identifiers, such as interface names, you must specify the identifier in a precise format. For example, the interface name so-0/0/0 refers to a SONET/SDH interface that is on the Flexible PIC Concentrator (FPC) in slot 0, in the first PIC location, and in the first port on the Physical Interface Card (PIC). For other identifiers, such as interface descriptive text and policy and firewall term names, you can specify any name, including special characters, spaces, and tabs.

You must enclose in quotation marks (double quotes) identifiers and any strings that include a space or tab character or any of the following characters:

```
( ) [ ] { } ! @ # $ % ^ & | ' = ?
```

If you do not type an option for a statement that requires one, a message indicates the type of information required. In this example, you need to type an area number to complete the command:

```
[edit]
user@host# set protocols ospf area<Enter>
                                  ^
syntax error, expecting <identifier>
```

Related Documentation

- Modifying the Junos OS Configuration on page 69
- Deleting a Statement from a Junos OS Configuration on page 71
- Copying a Junos OS Statement in the Configuration on page 73

Copyright © 2014, Juniper Networks, Inc.

- Renaming an Identifier in a Junos OS Configuration on page 75
- Using the configure exclusive Command on page 67
- Additional Details About Specifying Junos OS Statements and Identifiers on page 126
- Displaying the Current Junos OS Configuration on page 87

Deleting a Statement from a Junos OS Configuration

To delete a statement or identifier from a Junos OS configuration, use the **delete** configuration mode command. Deleting a statement or an identifier effectively "unconfigures" the functionality associated with that statement or identifier, returning that functionality to its default condition.

user@host# delete <*statement-path*> <*identifier*>

When you delete a statement, the statement and all its subordinate statements and identifiers are removed from the configuration.

For statements that can have more than one identifier, when you delete one identifier, only that identifier is deleted. The other identifiers in the statement remain.

To delete the entire hierarchy starting at the current hierarchy level, do not specify a statement or an identifier in the **delete** command. When you omit the statement or identifier, you are prompted to confirm the deletion:

```
[edit]
user@host# delete
Delete everything under this level? [yes, no] (no)
Possible completions:
no    Don't delete everything under this level
yes   Delete everything under this level
Delete everything under this level? [yes, no] (no)
```

Copyright © 2014, Juniper Networks, Inc.

NOTE: You cannot delete multiple statements or identifiers within a hierarchy using a single delete command. You must delete each statement or identifier individually using multiple delete commands. For example, consider the following configuration at the [edit system] hierarchy level:

```
system {
    host-name host-211;
    domain-name domain-122;
    backup-router 192.168.71.254;
    arp;
    authentication-order [ radius password tacplus ];
}
```

To delete the domain-name, host-name, and backup-router from the configuration, you cannot issue a single delete command:

```
user@host> delete system hostname host-211 domain-name domain-122 backup-router
192.168.71.254
```

You can only delete each statement individually:

```
user@host delete system host-name host-211
user@host delete system domain-name domain-122
user@host delete system backup-router 192.168.71.254
```

Related Documentation

- Example: Deleting a Statement from the Junos OS Configuration on page 72
- Adding Junos OS Configuration Statements and Identifiers on page 69
- Copying a Junos OS Statement in the Configuration on page 73

Example: Deleting a Statement from the Junos OS Configuration

The following example shows how to delete the **ospf** statement, effectively unconfiguring OSPF on the router:

```
[edit]
user@host# set protocols ospf area 0.0.0.0 interface so-0/0/0 hello-interval 5
[edit]
user@host# show
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0 {
                hello-interval 5;
            }
        }
    }
}
[edit]
user@host# delete protocols ospf
[edit]
user@host# show
[edit]
```


Copyright © 2014, Juniper Networks, Inc.

```
user@host#
```

Delete all statements from the current level down:

```
[edit]
user@host# edit protocols ospf area 0.0.0.0
[edit protocols ospf area 0.0.0.0]
user@host# set interface so-0/0/0 hello-interval 5
[edit protocols ospf area 0.0.0.0]
user@host# delete
Delete everything under this level? [yes, no] (no) yes
[edit protocols ospf area 0.0.0.0]
user@host# show
[edit]
user@host#
```

Unconfigure a particular property:

```
[edit]
user@host# set interfaces so-3/0/0 speed 100mb
[edit]
user@host# show
interfaces {
   so-3/0/0 {
      speed 100mb;
   }
}
[edit]
user@host# delete interfaces so-3/0/0 speed
[edit]
user@host# show
interfaces {
   so-3/0/0;
}
```

- Example: Using Global Replace in a Junos OS Configuration—Using the upto Option on page 192
- Deleting a Statement from a Junos OS Configuration on page 71

Copying a Junos OS Statement in the Configuration

When you have many similar statements in a Junos configuration, you can add one statement and then make copies of that statement. Copying a statement duplicates that statement and the entire hierarchy of statements configured under that statement. Copying statements is useful when you are configuring many physical or logical interfaces of the same type.

To make a copy of an existing statement in the configuration, use the configuration mode **copy** command:

```
user@host# copy existing-statement to new-statement
```

Immediately after you have copied a portion of the configuration, the configuration might not be valid. You must check the validity of the new configuration, and if necessary, modify either the copied portion or the original portion for the configuration to be valid.

Copyright © 2014, Juniper Networks, Inc.

Related Documentation
- Example: Copying a Statement in the Junos OS Configuration on page 74
- Adding Junos OS Configuration Statements and Identifiers on page 69

Example: Copying a Statement in the Junos OS Configuration

The following example shows how you can create one virtual connection (VC) on an interface, and then copy its configuration to create a second VC:

```
[edit interfaces]
user@host# show
at-1/0/0 {
    description "PAIX to MAE West"
    encapsulation atm-pvc;
    unit 61 {
        point-to-point;
        vci 0.61;
        family inet {
            address 10.0.1.1/24;
        }
    }
}
[edit interfaces]
user@host# edit at-1/0/0
[edit interfaces at-1/0/0]
user@host# copy unit 61 to unit 62
[edit interfaces at-1/0/0]
user@host# show
description "PAIX to MAE West"
encapsulation atm-pvc;
unit 61 {
    point-to-point;
    vci 0.61;
    family inet {
        address 10.0.1.1/24;
    }
}
unit 62 {
    point-to-point;
    vci 0.61;
    family inet {
        address 10.0.1.1/24;
    }
}
```

Related Documentation
- Copying a Junos OS Statement in the Configuration on page 73

Issuing Relative Junos OS Configuration Mode Commands

The **top** or **up** command followed by another configuration command, including **edit**, **insert**, **delete**, **deactivate**, **annotate**, or **show** enables you to quickly move to the top of the hierarchy or to a level above the area you are configuring.

Copyright © 2014, Juniper Networks, Inc.

To issue configuration mode commands from the top of the hierarchy, use the **top** command; then specify a configuration command. For example:

```
[edit interfaces fxp0 unit 0 family inet]
user@host# top edit system login
[edit system login]
user@host#
```

To issue configuration mode commands from a location higher up in the hierarchy, use the **up** configuration mode command; specify the number of levels you want to move up the hierarchy and then specify a configuration command. For example:

```
[edit protocols bgp]
user@host# up 2 activate system
```

Related Documentation

- Displaying the Current Junos OS Configuration on page 87

Renaming an Identifier in a Junos OS Configuration

When modifying a Junos configuration, you can rename an identifier that is already in the configuration. You can do this either by deleting the identifier (using the **delete** command) and then adding the renamed identifier (using the **set** and **edit** commands), or you can rename the identifier using the **rename** configuration mode command:

```
user@host# rename <statement-path> identifier1 to identifier2
```

Related Documentation

- Adding Junos OS Configuration Statements and Identifiers on page 69
- Example: Renaming an Identifier in a Junos OS Configuration on page 75
- Inserting a New Identifier in a Junos OS Configuration on page 75

Example: Renaming an Identifier in a Junos OS Configuration

This example shows how you can change the Network Time Protocol (NTP) server address to **10.0.0.6** using the **rename** configuration mode command:

```
[edit]
user@host# rename system network-time server 10.0.0.7 to server 10.0.0.6
```

Related Documentation

- Renaming an Identifier in a Junos OS Configuration on page 75

Inserting a New Identifier in a Junos OS Configuration

When configuring a device running Junos OS, you can enter most statements and identifiers in any order. Regardless of the order in which you enter the configuration statements, the CLI always displays the configuration in a strict order. However, there are a few cases where the ordering of the statements matters because the configuration statements create a sequence that is analyzed in order.

Copyright © 2014, Juniper Networks, Inc.

For example, in a routing policy or firewall filter, you define terms that are analyzed sequentially. Also, when you create a named path in dynamic MPLS, you define an ordered list of the transit routers in the path, starting with the first transit router and ending with the last one.

To modify a portion of the configuration in which the statement order matters, use the **insert** configuration mode command:

user@host# insert <*statement-path*> *identifier1* (before | after) *identifier2*

If you do not use the **insert** command, but instead simply configure the identifier, it is placed at the end of the list of similar identifiers.

Related Documentation

- Renaming an Identifier in a Junos OS Configuration on page 75
- Example: Renaming an Identifier in a Junos OS Configuration on page 75
- Example: Inserting a New Identifier in a Junos OS Configuration on page 76
- Deactivating and Reactivating Statements and Identifiers in a Junos OS Configuration on page 82

Example: Inserting a New Identifier in a Junos OS Configuration

Insert policy terms in a routing policy configuration. Note that if you do not use the **insert** command, but rather just configure another term, the added term is placed at the end of the existing list of terms. Also note that you must create the term, as shown in this example, before you can place it with the **insert** command.

```
[edit]
user@host# show
policy-options {
  policy-statement statics {
    term term1 {
      from {
        route-filter 192.168.0.0/16 orlonger;
        route-filter 224.0.0.0/3 orlonger;
      }
      then reject;
    }
    term term2 {
      from protocol direct;
      then reject;
    }
    term term3 {
      from protocol static;
      then reject;
    }
    term term4 {
      then accept;
    }
  }
}
[edit]
```


Copyright © 2014, Juniper Networks, Inc.

```
user@host# rename policy-options policy-statement statics term term4 to term term6
[edit]
user@host# set policy-options policy-statement statics term term4 from protocol local
[edit]
user@host# set policy-options policy-statement statics term term4 then reject
[edit]
user@host# set policy-options policy-statement statics term term5 from protocol
  aggregate
[edit]
user@host# set policy-options policy-statement statics term term5 then reject
[edit]
user@host# insert policy-options policy-statement statics term term4 after term term3
[edit]
user@host# insert policy-options policy-statement statics term term5 after term term4
[edit]
user@host# show policy-options policy-statement statics
term term1 {
  from {
    route-filter 192.168.0.0/16 orlonger;
    route-filter 224.0.0.0/3 orlonger;
  }
  then reject;
}
term term2 {
  from protocol direct;
  then reject;
}
term term3 {
  from protocol static;
  then accept;
}
term term4 {
  from protocol local;
  then reject;
}
term term5 {
  from protocol aggregate;
  then reject;
}
term term6 {
  then accept;
}
```

Insert a transit router in a dynamic MPLS path:

```
[edit protocols mpls path ny-sf]
user@host# show
1.1.1.1;
2.2.2.2;
3.3.3.3 loose;
4.4.4.4 strict;
6.6.6.6;
[edit protocols mpls path ny-sf]
user@host# insert 5.5.5.5 before 6.6.6.6
[edit protocols mpls path ny-sf]
user@host# set 5.5.5.5 strict
```

Copyright © 2014, Juniper Networks, Inc.

```
[edit protocols mpls path ny-sf]
user@host# show
1.1.1.1;
2.2.2.2;
3.3.3.3 loose;
4.4.4.4 strict;
5.5.5.5 strict;
6.6.6.6;
```

Related Documentation

- Inserting a New Identifier in a Junos OS Configuration on page 75
- Adding Junos OS Configuration Statements and Identifiers on page 69

Example: Using the Wildcard Command with the Range Option

- Requirements on page 78
- Overview on page 78
- Configuration on page 79
- Verification on page 81

Requirements

This example uses the following hardware and software components:

- M Series, MX Series, T Series or EX Series device
- Junos OS Release 12.1 or later running on the device

Overview

The **range** option with the **wildcard** command enables you to specify ranges in **activate**, **deactivate**, **delete**, **protect**, **set**, **show**, and **unprotect** commands. You can use ranges to specify a range of interfaces, logical units, VLANs, and other numbered elements. The **wildcard range** option expands the command you entered into multiple commands, each of which corresponds to one item in the range.

The **wildcard range** option enables you to configure multiple configuration statements using a single **set** command, instead of configuring each of them individually. For example, to configure 24 Gigabit Ethernet interfaces with different port numbers, you can use a single **wildcard range set** command instead of 24 individual **set interfaces** commands.

Similarly, to deactivate a group of 30 logical interfaces, you can use the **wildcard range deactivate** command instead of deactivating each logical interface individually.

You can use **wildcard range** with the **active**, **deactivate**, **delete**, **protect**, **set**, **show**, and **unprotect** configuration commands:

```
user@host# wildcard range ?
Possible completions:
   activate             Remove the inactive tag from a statement
   deactivate           Add the inactive tag to a statement
   delete               Delete a data element
   protect              Protect the statement
```


Copyright © 2014, Juniper Networks, Inc.

```
set                 Set a parameter
show                Show a parameter
unprotect           Unprotect the statement
```

You can also specify all configuration hierarchy levels and their child configuration statements in the CLI by using **wildcard range** with the **set** option:

```
Possible completions:
> >  access               Network access configuration
> >  access-profile       Access profile for this instance
> >  accounting-options   Accounting data configuration
> >  applications         Define applications by protocol characteristics
...
```

Configuration

The following examples show how to configure multiple configuration statements in a single step by using the **range** option with the **wildcard** configuration command:

- Using the Range Option for Configuring a Series of Named Identifiers for a Configuration Statement on page 79
- Specifying Multiple Ranges in the Syntax on page 80
- Specifying a Range and Unique Numbers In the Syntax on page 80
- Excluding Some Values from a Range on page 80
- Specifying a Range with a Step Number on page 81

Using the Range Option for Configuring a Series of Named Identifiers for a Configuration Statement

Step-by-Step Procedure

You can configure a series of identifiers for a configuration statement, by specifying a numerical range of values for the identifiers.

- To configure a series of the same type of interface with different port numbers (**0** through **23**), specify the range for the port numbers by using the following format:

  ```
  [edit]
  user@host# wildcard range set interfaces ge-0/0/[0-23] unit 0 family vpls
  ```

Results

Expands to 24 different **set** commands to configure interfaces with port numbers ranging from **0** through **23**:

```
[edit]
user@host# set interfaces ge-0/0/0 unit 0 family vpls
user@host# set interfaces ge-0/0/1 unit 0 family vpls
user@host# set interfaces ge-0/0/2 unit 0 family vpls
...
user@host# set interfaces ge-0/0/23 unit 0 family vpls
```

Copyright © 2014, Juniper Networks, Inc.

Specifying Multiple Ranges in the Syntax

Step-by-Step Procedure

You can have multiple ranges specified in a **wildcard range** command. Each range must be separated by a comma. You can also have overlapping ranges.

- To specify more than one range in the syntax, include the minimum and maximum values for each range, separated by a comma.

```
[edit]
user@host# wildcard range protect event-options policy p[1-3,5-7,6-9]
```

Results

Expands to the following **set** commands:

```
[edit]
user@host# set protect event-options policy p1
user@host# set protect event-options policy p2
user@host# set protect event-options policy p3
user@host# set protect event-options policy p5
user@host# set protect event-options policy p6
user@host# set protect event-options policy p7
user@host# set protect event-options policy p8
user@host# set protect event-options policy p9
```

Specifying a Range and Unique Numbers In the Syntax

Step-by-Step Procedure

You can also specify a combination of a range and unique numbers in the syntax of the **wildcard range** command.

- To specify a range and unique numbers, separate them with a comma.

```
[edit]
user@host# wildcard range protect event-options policy p[1-3,5,7,10]
```

Results

Expands to the following **set** commands:

```
[edit]
user@host# set protect event-options policy p1
user@host# set protect event-options policy p2
user@host# set protect event-options policy p3
user@host# set protect event-options policy p5
user@host# set protect event-options policy p7
user@host# set protect event-options policy p10
```

Excluding Some Values from a Range

Step-by-Step Procedure

You can exclude certain values from a range by marking the numbers or the range of numbers to be excluded by using an exclamation mark.

- To exclude certain values from a range, include the portion to be excluded with **!** in the syntax.

```
[edit]
user@host# wildcard range protect event-options policy p[1-5,!3-4]
```


Copyright © 2014, Juniper Networks, Inc.

Results Expands to the following **set** commands:

```
[edit]
user@host# set protect event-options policy p1
user@host# set protect event-options policy p2
user@host# set protect event-options policy p5
```

Specifying a Range with a Step Number

Step-by-Step Procedure You can provide a step number for a range to have a constant interval in the range.

- To provide a step, include the step value in the syntax preceded by a forward slash (/).

```
[edit]
user@host# wildcard range protect event-options policy p[1-10/2]
```

Results Expands to the following **set** commands:

```
[edit]
user@host# set protect event-options policy p1
user@host# set protect event-options policy p3
user@host# set protect event-options policy p5
user@host# set protect event-options policy p7
user@host# set protect event-options policy p9
```

Verification

Confirm that the configuration is working properly.

- Checking the Configuration on page 81

Checking the Configuration

Purpose Check the configuration created using the **wildcard range** option. The following sample shows output for the configuration described in "Using the Range Option for Configuring a Series of Named Identifiers for a Configuration Statement" on page 79.

Copyright © 2014, Juniper Networks, Inc.

Action

```
user@host> show configuration interfaces
ge-0/0/0 {
    unit 0 {
        family vpls;
    }
}
ge-0/0/1 {
    unit 0 {
        family vpls;
    }
}
ge-0/0/2 {
    unit 0 {
        family vpls;
    }
}
ge-0/0/3 {
    unit 0 {
        family vpls;
    }
}
...
ge-0/0/23 {
    unit 0 {
        family vpls;
    }
}
```

Meaning The output indicates that 24 Gigabit Ethernet interfaces ranging from **ge-0/0/0** through **ge-0/0/23** are created.

Related Documentation

- Using Wildcard Characters in Interface Names on page 187

Deactivating and Reactivating Statements and Identifiers in a Junos OS Configuration

In a Junos configuration, you can deactivate statements and identifiers so that they do not take effect when you issue the **commit** command. Any deactivated statements and identifiers are marked with the **inactive:** tag. They remain in the configuration, but are not activated when you issue a **commit** command.

To deactivate a statement or identifier, use the **deactivate** configuration mode command:

user@host# **deactivate** (***statement identifier***)

To reactivate a statement or identifier, use the **activate** configuration mode command:

user@host# activate (***statement identifier***)

In both commands, the ***statement*** and ***identifier*** you specify must be at the current hierarchy level.

In some portions of the configuration hierarchy, you can include a **disable** statement to disable functionality. One example is disabling an interface by including the **disable** statement at the **[edit interface *interface-name*]** hierarchy level. When you deactivate a

Copyright © 2014, Juniper Networks, Inc.

statement, that specific object or property is completely ignored and is not applied at all when you issue a **commit** command. When you disable a functionality, it is activated when you issue a **commit** command but is treated as though it is down or administratively disabled.

Related Documentation

- Examples: Deactivating and Reactivating Statements and Identifiers in a Junos OS Configuration on page 83
- Adding Junos OS Configuration Statements and Identifiers on page 69

Examples: Deactivating and Reactivating Statements and Identifiers in a Junos OS Configuration

Deactivate an interface in the configuration:

```
[edit interfaces]
user@host# show
at-5/2/0 {
    traceoptions {
        traceflag all;
    }
    atm-options {
        vpi 0 maximum-vcs 256;
    }
    unit 0 {
        ...
        [edit interfaces]
        user@host# deactivate at-5/2/0
        [edit interfaces]
        user@host# show
        inactive: at-5/2/0 {
            traceoptions {
                traceflag all;
            }
            ...
        }
    }
}
```

Reactivate the interface:

```
[edit interfaces]
user@host# activate at-5/2/0
[edit interfaces]
user@host# show
at-5/2/0 {
    traceoptions {
        traceflag all;
    }
    ...
}
```

Related Documentation

- Deactivating and Reactivating Statements and Identifiers in a Junos OS Configuration on page 82

Copyright © 2014, Juniper Networks, Inc.

Adding Comments in a Junos OS Configuration

You can include comments in a Junos configuration to describe any statement in the configuration. You can add comments interactively in the CLI and by editing the ASCII configuration file.

When you add comments in configuration mode, they are associated with a statement at the current level. Each statement can have one single-line comment associated with it. Before you can associate a comment with a statement, the statement must exist. The comment is placed on the line preceding the statement.

To add comments to a configuration, use the **annotate** configuration mode command:

```
user@host# annotate statement "comment-string"
```

statement is the configuration statement to which you are attaching the comment; it must be at the current hierarchy level. If a comment for the specified ***statement*** already exists, it is deleted and replaced with the new comment.

comment-string is the text of the comment. The comment text can be any length, and you must type it on a single line. If the comment contains spaces, you must enclose it in quotation marks. In the comment string, you can include the comment delimiters **/* */** or **#**. If you do not specify any, the comment string is enclosed with the **/* */** comment delimiters.

To delete an existing comment, specify an empty comment string:

```
user@host# annotate statement ""
```

When you edit the ASCII configuration file and add comments, they can be one or more lines and must precede the statement they are associated with. If you place the comments in other places in the file, such as on the same line following a statement or on a separate line following a statement, they are removed when you use the **load** command to open the configuration into the CLI.

When you include comments in the configuration file directly, you can format comments in the following ways:

- Start the comment with a **/*** and end it with a ***/**. The comment text can be on a single line or can span multiple lines.
- Start the comment with a **#** and end it with a new line (carriage return).

If you add comments with the **annotate** command, you can view the comments within the configuration by entering the show configuration mode command or the **show configuration** operational mode command.

When configuring interfaces, you can add comments about the interface by including the **description** statement at the **[edit interfaces *interface-name*]** hierarchy level. Any comments you include appear in the output of the **show interfaces** commands. For more information about the **description** statement, see the *Junos OS Network Interfaces Library for Routing Devices*.

Copyright © 2014, Juniper Networks, Inc.

NOTE: The Junos OS supports annotation up to the last level in the configuration hierarchy, including oneliners. However, annotation of parts (the child statements or identifiers within the oneliner) of the oneliner is not supported. For example, in the following sample configuration hierarchy, annotation is supported up to the level 1 parent hierarchy, but not supported for the metric child statement:

```
[edit protocols]
   isis {
      interface ge-0/0/0.0 {
         level 1 metric 10;
      }
   }
}
```

Related Documentation

- Adding Junos OS Configuration Statements and Identifiers on page 69
- Example: Including Comments in a Junos OS Configuration on page 85

Example: Including Comments in a Junos OS Configuration

To add comments to a Junos configuration:

```
[edit]
user@host# show
protocols {
   ospf {
      area 0.0.0.0 {
         interface so-0/0/0 {
            hello-interval 5;
         }
      }
   }
}
[edit]
user@host# edit protocols ospf
[edit protocols ospf]
user@host# set area 0.0.0.0
user@host# annotate area 0.0.0.0 "Backbone area configuration added June 15, 1998"
[edit protocols ospf]
user@host# edit area 0.0.0.0
[edit protocols ospf area 0.0.0.0]
user@host# annotate interface so0 "Interface from router sj1 to router sj2"
[edit protocols ospf area 0.0.0.0]
user@host# top
[edit]
user@host# show
protocols {
   ospf {
      /* Backbone area configuration added June 15, 1998 */
      area 0.0.0.0 {
         /* Interface from router sj1 to router sj2 */
```

Copyright © 2014, Juniper Networks, Inc.

```
        interface so-0/0/0 {
          hello-interval 5;
        }
      }
    }
  }
  [edit]
  user@host#
```

The following excerpt from a configuration example illustrates how to enter comments in a configuration file:

```
/* This comment goes with routing-options */
routing-options {
  /* This comment goes with routing-options traceoptions */
  traceoptions {
    /* This comment goes with routing-options traceoptions tracefile */
    tracefile rpd size 1m files 10;
    /* This comment goes with routing-options traceoptions traceflag task */
    traceflag task;
    /* This comment goes with routing-options traceoptions traceflag general */
    traceflag general;
  }
  autonomous-system 10458; /* This comment is dropped */
}
routing-options {
  rib-groups {
    ifrg {
      import-rib [ inet.0 inet.2 ];
      /* A comment here is dropped */
    }
    dvmrp-rib {
      import-rib inet.2;
      export-rib inet.2;
      /* A comment here is dropped */
    }
    /* A comment here is dropped */
  }
  /* A comment here is dropped */
}
```

Related Documentation

- Adding Comments in a Junos OS Configuration on page 84

Updating the configure private Configuration

When you are in configure private mode, you must work with a copy of the most recently committed shared configuration. If the global configuration changes, you can issue the **update** command to update your private candidate configuration. When you do this, your private candidate configuration contains a copy of the most recently committed configuration with your private changes merged in. For example:

```
[edit]
user@host# update
[edit]
```


Copyright © 2014, Juniper Networks, Inc.

```
user@host#
```

NOTE: Merge conflicts can occur when you issue the update command.

You can also issue the **rollback** command to discard your private candidate configuration changes and obtain the most recently committed configuration:

```
[edit]
user@host# rollback
[edit]
user@host#
```

Related Documentation

- Forms of the configure Command on page 66

Displaying the Current Junos OS Configuration

To display the current configuration for a device running Junos OS, use the **show** configuration mode command. This command displays the configuration at the current hierarchy level or at the specified level.

```
user@host# show <statement-path>
```

The configuration statements appear in a fixed order, interfaces appear alphabetically by type, and then in numerical order by slot number, PIC number, and port number. Note that when you configure the router, you can enter statements in any order.

You also can use the CLI operational mode **show configuration** command to display the last committed current configuration, which is the configuration currently running on the router:

```
user@host> show configuration
```

When you show a configuration, a timestamp at the top of the configuration indicates when the configuration was last changed:

```
## Last commit: 2006-07-18 11:21:58 PDT by echen
version 8.3
```

If you have omitted a required statement at a particular hierarchy level, when you issue the **show** command in configuration mode, a message indicates which statement is missing. As long as a mandatory statement is missing, the CLI continues to display this message each time you issue a **show** command. For example:

```
[edit]
user@host# show
protocols {
   pim {
      interface so-0/0/0 {
         priority 4;
         version 2;
         # Warning: missing mandatory statement(s): 'mode'
```

Copyright © 2014, Juniper Networks, Inc.

```
    }
  }
}
```

When you issue the **show configuration** command with the **| display set** pipe option to view the configuration as **set** commands, those portions of the configuration that you do not have permissions to view are substituted with the text **ACCESS-DENIED**.

Unsupported statements included in the CLI configuration are displayed with the "unsupported" text in the configuration. For example, if a statement is configured on an unsupported platform, the CLI displays a message that the statement is ignored in the configuration because it is configured on an unsupported platform. When you issue the **show** command with the **| display xml** option, you can see the **unsupported="unsupported"** attribute for configuration that is unsupported.

The "unsupported" attribute included in text configuration or XML configuration is provided to scripts when the **unsupported="unsupported"** attribute is included in the **<get-configuration>** RPC call.

Related Documentation

- Example: Displaying the Current Junos OS Configuration on page 88
- Displaying set Commands from the Junos OS Configuration on page 92

Example: Displaying the Current Junos OS Configuration

The following example shows how you can display the current Junos configuration. To display the entire configuration:

```
[edit]
user@host# set protocols ospf area 0.0.0.0 interface so-0/0/0 hello-interval 5
[edit]
user@host# show
protocols {
  ospf {
    area 0.0.0.0 {
      interface so-0/0/0 {
        hello-interval 5;
      }
    }
  }
}
```

Display a particular hierarchy in the configuration:

```
[edit]
user@host# show protocols ospf area 0.0.0.0
interface so-0/0/0 {
  hello-interval 5;
}
```

Move down a level and display the configuration at that level:

```
[edit]
user@host# edit protocols ospf area 0.0.0.0
[edit protocols ospf area 0.0.0.0]
```


Copyright © 2014, Juniper Networks, Inc.

```
user@host# show
interface so-0/0/0 {
  hello-interval 5;
}
```

Display all of the last committed configuration:

```
[edit]
user@host# set protocols ospf area 0.0.0.0 interface so-0/0/0 hello-interval 5
[edit]
user@host# commit
commit complete
[edit]
user@host# quit
exiting configuration mode
user@host> show configuration
## Last commit: 2006-08-10 11:21:58 PDT by user
version 8.3
protocols {
  ospf {
    area 0.0.0.0 {
      interface so-0/0/0 {
        hello-interval 5;
      }
    }
  }
}
```

Related Documentation

- Displaying the Current Junos OS Configuration on page 87

Displaying Additional Information About the Junos OS Configuration

In configuration mode only, to display additional information about the configuration, use the **display detail** command after the pipe (|) in conjunction with a **show** command. The additional information includes the help string that explains each configuration statement and the permission bits required to add and modify the configuration statement.

```
user@host# show <hierarchy-level> | display detail
```

For example:

```
[edit]
user@host# show | display detail
##
## version: Software version information
## require: system
##
version "3.4R1 [tlim]";
system {
  ##
  ## host-name: Host name for this router
  ## match: ^[[:alnum:]._-]+$
  ## require: system
  ##
```

Copyright © 2014, Juniper Networks, Inc.

```
}
host-name router-name;
##
## domain-name: Domain name for this router
## match: ^[[:alnum:]._-]+$
## require: system
##
domain-name isp.net;
##
## backup-router: Address of router to use while booting
##
backup-router 192.168.100.1;
root-authentication {
    ##
    ## encrypted-password: Encrypted password string
    ##
    encrypted-password "$1$BYJQE$/ocQof8pmcm7MSGK0"; # SECRET-DATA
}
##
## name-server: DNS name servers
## require: system
##
name-server {
    ##
    ## name-server: DNS name server address
    ##
    208.197.1.0;
}
login {
    ##
    ## class: User name (login)
    ## match: ^[[:alnum:]_-]+$
    ##
    class super-user {
        ##
        ## permissions: Set of permitted operation categories
        ##
        permissions all;
    }
    ...
    ##
    ## services: System services
    ## require: system
    ##
    services {
        ## services: Service name
        ##
        ftp;
        ##
        ## services: Service name
        ##
        telnet;
        ##
    }
    syslog {
        ##
```


Copyright © 2014, Juniper Networks, Inc.

```
      ## file-name: File to record logging data
      ##
      file messages {
        ##
        ## Facility type
        ## Level name
        ##
        any notice;
        ##
        ## Facility type
        ## Level name
        ##
        authorization info;
      }
    }
}
chassis {
    alarm {
      sonet {
        ##
        ## lol: Loss of light
        ## alias: loss-of-light
        ##
        lol red;
      }
    }
}
interfaces {
    ##
    ## Interface name
    ##
    at-2/1/1 {
      atm-options {
        ##
        ## vpi: Virtual path index
        ## range: 0 .. 255
        ## maximum-vcs: Maximum number of virtual circuits on this VP
        ##
        vpi 0 maximum-vcs 512;
      }
      ##
      ## unit: Logical unit number
      ## range: 0 .. 16384
      ##
      unit 0 {
        ##
        ## vci: ATM point-to-point virtual circuit identifier ([vpi.]vci)
      }
      ##
      vci 0.128;
    }
}
...
```

Copyright © 2014, Juniper Networks, Inc.

Related Documentation

- Displaying set Commands from the Junos OS Configuration on page 92

Displaying set Commands from the Junos OS Configuration

In configuration mode, you can display the configuration as a series of configuration mode commands required to re-create the configuration. This is useful if you are not familiar with how to use configuration mode commands or if you want to cut, paste, and edit the displayed configuration.

To display the configuration as a series of configuration mode commands, which are required to re-create the configuration from the top level of the hierarchy as **set** commands, issue the **show** configuration mode command with the **display set** option:

```
user@host# show | display set
```

This topic contains the following examples:

- Example: Displaying set Commands from the Configuration on page 92
- Example: Displaying Required set Commands at the Current Hierarchy Level on page 93
- Example: Displaying set Commands with the match Option on page 93

Example: Displaying set Commands from the Configuration

Display the **set** commands from the configuration at the [**edit interfaces**] hierarchy level:

```
[edit interfaces fe-0/0/0]
user@host# show
unit 0 {
    family inet {
        address 192.107.1.230/24;
    }
    family iso;
    family mpls;
}
inactive: unit 1 {
    family inet {
        address 10.0.0.1/8;
    }
}
user@host# show | display set
set interfaces fe-0/0/0 unit 0 family inet address 192.107.1.230/24
set interfaces fe-0/0/0 unit 0 family iso
set interfaces fe-0/0/0 unit 0 family mpls
set interfaces fe-0/0/0 unit 1 family inet address 10.0.0.1/8
deactivate interfaces fe-0/0/0 unit 1
```

To display the configuration as a series of configuration mode commands required to re-create the configuration from the current hierarchy level, issue the **show** configuration mode command with the **display set relative** option:

```
user@host# show | display set relative
```


Copyright © 2014, Juniper Networks, Inc.

Example: Displaying Required set Commands at the Current Hierarchy Level

Display the configuration as a series of configuration mode commands required to re-create the configuration from the current hierarchy level:

```
[edit interfaces fe-0/0/0]
user@host# show
unit 0 {
    family inet {
        address 192.107.1.230/24;
    }
    family iso;
    family mpls;
}
inactive: unit 1 {
    family inet {
        address 10.0.0.1/8;
    }
}
user@host# show | display set relative
set unit 0 family inet address 192.107.1.230/24
set unit 0 family iso
set unit 0 family mpls
set unit 1 family inet address 10.0.0.1/8
deactivate unit 1
```

To display the configuration as **set** commands and search for text matching a regular expression by filtering output, specify the **match** option after the pipe (|):

```
user@host# show | display set | match regular-expression
```

Example: Displaying set Commands with the match Option

Display IP addresses associated with an interface:

```
xe-2/3/0 {
    unit 0 {
        family inet {
            address 192.107.9.106/30;
        }
    }
}
so-5/1/0 {
    unit 0 {
        family inet {
            address 192.107.9.15/32 {
                destination 192.107.9.192;
            }
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 127.0.0.1/32;
```

Copyright © 2014, Juniper Networks, Inc.

```
    }
  }
}
user@host# show interfaces | display set | match address
set interfaces xe-2/3/0 unit 0 family inet address 192.168.9.106/30
set interfaces so-5/1/0 unit 0 family inet address 192.168.9.15/32 destination 192.168.9.192
set interfaces lo0 unit 0 family inet address 127.0.0.1/32
```

Related Documentation
- Displaying the Current Junos OS Configuration on page 87

Displaying Users Currently Editing the Junos OS Configuration

To display the users currently editing the configuration, use the **status** configuration mode command:

```
user@host# status
Users currently editing the configuration:
rchen terminal p0 (pid 55691) on since 2006-03-01 13:17:25 PST
[edit interfaces]
```

The system displays who is editing the configuration (**rchen**), where the user is logged in (**terminal p0**), the date and time the user logged in (**2006-03-01 13:17:25 PST**), and what level of the hierarchy the user is editing (**[edit interfaces]**).

If you issue the **status** configuration mode command and a user has scheduled a candidate configuration to become active for a future time, the system displays who scheduled the commit (**root**), where the user is logged in (**terminal d0**), the date and time the user logged in (**2002-10-31 14:55:15 PST**), and that a commit is pending **(commit at).**

```
[edit]
user@host# status
Users currently editing the configuration:
root terminal d0 (pid 767) on since 2002-10-31 14:55:15 PST, idle 00:03:09
commit at
```

For information about how to schedule a commit, see "Scheduling a Junos OS Commit Operation" on page 103.

If you issue the **status** configuration mode command and a user is editing the configuration in configure exclusive mode, the system displays who is editing the configuration (**root**), where the user is logged in (**terminal d0**), the date and time the user logged in (**2002-11-01 13:05:11 PST**), and that a user is editing the configuration in configure exclusive mode (**exclusive [edit]**).

```
[edit]
user@host# status
Users currently editing the configuration:
root terminal d0 (pid 2088) on since 2002-11-01 13:05:11 PST
exclusive [edit]
```

Related Documentation
- Forms of the configure Command on page 66
- Using the configure exclusive Command on page 67

Copyright © 2014, Juniper Networks, Inc.

Verifying a Junos OS Configuration

To verify that the syntax of a Junos configuration is correct, use the configuration mode **commit check** command:

```
[edit]
user@host# commit check
configuration check succeeds
[edit]
user@host#
```

If the **commit check** command finds an error, a message indicates the location of the error.

Related Documentation

- Adding Junos OS Configuration Statements and Identifiers on page 69
- Committing a Junos OS Configuration on page 98

Copyright © 2014, Juniper Networks, Inc.

Copyright © 2014, Juniper Networks, Inc.

CHAPTER 5

Committing a Junos OS Configuration

- Junos OS Commit Model for Router or Switch Configuration on page 97
- Committing a Junos OS Configuration on page 98
- Committing a Junos OS Configuration and Exiting Configuration Mode on page 100
- Commit Operation When Multiple Users Configure the Software on page 101
- Activating a Junos OS Configuration but Requiring Confirmation on page 102
- Scheduling a Junos OS Commit Operation on page 103
- Monitoring the Junos OS Commit Process on page 104
- Adding a Comment to Describe the Committed Configuration on page 105
- Backing Up the Committed Configuration on the Alternate Boot Drive on page 106
- Junos OS Batch Commits Overview on page 107
- Example: Configuring Batch Commit Server Properties on page 107

Junos OS Commit Model for Router or Switch Configuration

The router or switch configuration is saved using a commit model—a candidate configuration is modified as desired and then committed to the system. When a configuration is committed, the router or switch checks the configuration for syntax errors, and if no errors are found, the configuration is saved as **juniper.conf.gz** and activated. The formerly active configuration file is saved as the first rollback configuration file (**juniper.conf.1.gz**), and any other rollback configuration files are incremented by 1. For example, **juniper.conf.1.gz** is incremented to **juniper.conf.2.gz**, making it the second rollback configuration file. The router or switch can have a maximum of 49 rollback configurations (numbered 1 through 49) saved on the system.

On the router or switch, the active configuration file and the first three rollback files (**juniper.conf.gz.1, juniper.conf.gz.2, juniper.conf.gz.3**) are located in the **/config** directory. If the file **rescue.conf.gz** is saved on the system, this file should also be saved in the **/config** directory. The factory default files are located in the **/etc/config** directory.

There are two mechanisms used to propagate the configurations between Routing Engines within a router or switch:

- Synchronization—Propagates a configuration from one Routing Engine to a second Routing Engine within the same router or switch chassis.

Copyright © 2014, Juniper Networks, Inc.

NOTE: The QFX3500 switch has only one Routing Engine.

To synchronize configurations, use the **commit synchronize** CLI command. If one of the Routing Engines is locked, the synchronization fails. If synchronization fails because of a locked configuration file, you can use the **commit synchronize force** command. This command overrides the lock and synchronizes the configuration files.

- Distribution—Propagates a configuration across the routing plane on a multichassis router or switch. Distribution occurs automatically. There is no user command available to control the distribution process. If a configuration is locked during a distribution of a configuration, the locked configuration does not receive the distributed configuration file, so the synchronization fails. You need to clear the lock before the configuration and resynchronize the routing planes.

NOTE: When you use the **commit synchronize force** CLI command on a multichassis platform, the forced synchronization of the configuration files does not affect the distribution of the configuration file across the routing plane. If a configuration file is locked on a router or switch remote from the router or switch where the command was issued, the synchronization fails on the remote router or switch. You need to clear the lock and reissue the **synchronization** command.

Related Documentation

- *Configuring Junos OS for the First Time on a Router or Switch with a Single Routing Engine*
- commit on page 240

Committing a Junos OS Configuration

To save Junos OS configuration changes to the configuration database and to activate the configuration on the router, use the **commit** configuration mode command. You can issue the **commit** command from any hierarchy level:

```
[edit]
user@host# commit
commit complete
[edit]
user@host#
```

When you enter the **commit** command, the configuration is first checked for syntax errors (**commit check**). Then, if the syntax is correct, the configuration is activated and becomes the current, operational router configuration.

You can issue the **commit** command from any hierarchy level.

A configuration commit can fail for any of the following reasons:

- The configuration includes incorrect syntax, which causes the commit check to fail.

Copyright © 2014, Juniper Networks, Inc.

- The candidate configuration that you are trying to commit is larger than 700 MB.
- The configuration is locked by a user who entered the **configure exclusive** command.

If the configuration contains syntax errors, a message indicates the location of the error, and the configuration is not activated. The error message has the following format:

```
[edit edit-path]
'offending-statement;'
error-message
```

For example:

```
[edit firewall filter login-allowed term allowed from]
'icmp-type [ echo-request echo-reply ];'
keyword 'echo-reply' unrecognized
```

You must correct the error before recommitting the configuration. To return quickly to the hierarchy level where the error is located, copy the path from the first line of the error and paste it at the configuration mode prompt at the **[edit]** hierarchy level.

The uncommitted, candidate configuration file is **/var/rundb/juniper.db**. It is limited to 700 MB. If the commit fails with a message **configuration database size limit exceeded**, view the file size from configuration mode by entering the command **run file list /var/rundb detail**. You can simplify the configuration and reduce the file size by creating configuration groups with wildcards or defining less specific match policies in your firewall filters.

NOTE: CLI commit-time warnings displayed for configuration changes at the **[edit interfaces]** hierarchy level are removed and are logged as system log messages.

This is also applicable to VRRP configuration at the following hierarchy levels:

- **[edit interfaces** *interface-name* **unit** *logical-unit-number* **family (***inet* **|** *inet6***) address** *address***]**
- **[edit logical-systems** *logical-system-name* **interfaces** *interface-name* **unit** *logical-unit-number* **family (inet | inet6) address** *address***]**

When you commit a configuration, you commit the entire configuration in its current form. If more than one user is modifying the configuration, committing it saves and activates the changes of all the users.

Copyright © 2014, Juniper Networks, Inc.

NOTE:

- If you are using Junos OS in a Common Criteria environment, system log messages are created whenever a secret attribute is changed (for example, password changes or changes to the RADIUS shared secret). These changes are logged during the following configuration load operations:

  load merge
  load replace
  load override
  load update

  For more information, see the *Secure Configuration Guide for Common Criteria and Junos-FIPS*.

- We do not recommend performing a commit operation on the backup Routing Engine when graceful Routing Engine switchover is enabled on the router.

- NOTE: If you configure the same IP address for a management interface or internal interface such as fxp0 and an external physical interface such as ge-0/0/1, when graceful Routing Engine switchover (GRES) is enabled, the CLI displays an appropriate commit error message that identical addresses have been found on the private and public interfaces. In such cases, you must assign unique IP addresses for the two interfaces that have duplicate addresses.

  The management Ethernet interface used for the TX Matrix Plus router, T1600 or T4000 routers in a routing matrix, and PTX Series Packet Transport Routers, is em0. Junos OS automatically creates the router's management Ethernet interface, em0.

Related Documentation

- Committing a Junos OS Configuration and Exiting Configuration Mode on page 100
- Activating a Junos OS Configuration but Requiring Confirmation on page 102
- Backing Up the Committed Configuration on the Alternate Boot Drive on page 106
- Forms of the configure Command on page 66

Committing a Junos OS Configuration and Exiting Configuration Mode

To save Junos OS configuration changes, activate the configuration on the device and exit configuration mode, using the **commit and-quit** configuration mode command. This command succeeds only if the configuration contains no errors.

```
[edit]
user@host# commit and-quit
commit complete
```


Copyright © 2014, Juniper Networks, Inc.

```
exiting configuration mode
user@host>
```

NOTE: We do not recommend performing a commit operation on the backup Routing Engine when graceful Routing Engine switchover is enabled on the router.

Related Documentation

- Activating a Junos OS Configuration but Requiring Confirmation on page 102

Commit Operation When Multiple Users Configure the Software

Up to 32 users can be in configuration mode simultaneously, and they all can be making changes to the configuration. All changes made by all users are visible to everyone editing the configuration—the changes become visible as soon as the user presses the Enter key at the end of a command that changes the configuration, such as **set**, **edit**, or **delete**.

When any of the users editing the configuration issues a **commit** command, all changes made by all users are checked and activated.

If you enter configuration mode with the **configure private** command, each user has a private candidate configuration to edit somewhat independently of other users. When you commit the configuration, only your own changes get committed. To synchronize your copy of the configuration after other users have committed changes, you can run the **update** command in configuration mode. A commit operation also updates all of the private candidate configurations. For example, suppose user X and user Y are both in **configure private** mode, and user X commits a configuration change. When user Y performs a subsequent commit operation and then views the new configuration, the new configuration seen by user Y includes the changes made by user X.

If you enter configuration mode with the **configure exclusive** command, you lock the candidate configuration for as long as you remain in configuration mode, allowing you to make changes without interference from other users. Other users can enter and exit configuration mode, but they cannot commit the configuration. This is true even if the other users entered configuration mode before you enter the **configure exclusive** command. For example, suppose user X is already in the **configure private** or **configure** mode. Then suppose user Y enters the **configure exclusive** mode. User X cannot commit any changes to the configuration, even if those changes were entered before user Y logged in. If user Y exits **configure exclusive** mode, user X can then commit the changes made in **configure private** or **configure** mode.

Related Documentation

- Committing a Junos OS Configuration on page 98
- Forms of the configure Command on page 66
- Displaying Users Currently Editing the Junos OS Configuration on page 94

Copyright © 2014, Juniper Networks, Inc.

Activating a Junos OS Configuration but Requiring Confirmation

When you commit the current candidate configuration, you can require an explicit confirmation for the commit to become permanent. This is useful if you want to verify that a configuration change works correctly and does not prevent access to the router. If the change prevents access or causes other errors, the router automatically returns to the previous configuration and restores access after the rollback confirmation timeout passes. This feature is called automatic rollback.

To commit the current candidate configuration but require an explicit confirmation for the commit to become permanent, use the **commit confirmed** configuration mode command:

```
[edit]
user@host# commit confirmed
commit confirmed will be automatically rolled back in 10 minutes unless confirmed
commit complete
#commit confirmed will be rolled back in 10 minutes
[edit]
user@host#
```

Once you have verified that the change works correctly, you can keep the new configuration active by entering a **commit** or **commit check** command within 10 minutes of the **commit confirmed** command. For example:

```
[edit]
user@host# commit check
commit confirmed will be automatically rolled back in 10 minutes unless confirmed
commit complete
#commit confirmed will be rolled back in 10 minutes
[edit]
user@host#
```

If the commit is not confirmed within a certain time (10 minutes by default), Junos OS automatically rolls back to the previous configuration and a broadcast message is sent to all logged-in users.

To show when a rollback is scheduled after a **commit confirmed** command, enter the **show system commit** command. For example:

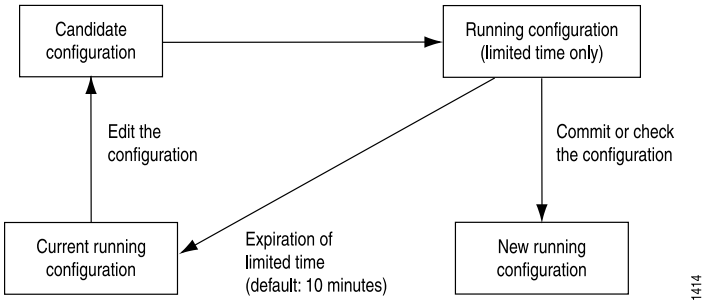
```
user@host>show system commit
0 2005-01-05 15:00:37 PST by root via cli commit confirmed, rollback in 3mins
```

Like the **commit** command, the **commit confirmed** command verifies the configuration syntax and reports any errors. If there are no errors, the configuration is activated and begins running on the router.

Figure 5 on page 103 illustrates how the **commit confirmed** command works.

 Copyright © 2014, Juniper Networks, Inc.

Figure 5: Confirm a Configuration



To change the amount of time before you have to confirm the new configuration, specify the number of minutes when you issue the command:

```
[edit]
user@host# commit confirmed minutes
commit complete
[edit]
user@host#
```

In Junos OS Release 11.4 and later, you can also use the **commit confirmed** command in the **[edit private]** configuration mode.

Related Documentation

- Scheduling a Junos OS Commit Operation on page 103
- Committing a Junos OS Configuration on page 98

Scheduling a Junos OS Commit Operation

You can schedule when you want your candidate configuration to become active. To save Junos OS configuration changes and activate the configuration on the router at a future time or upon reboot, use the **commit at** configuration mode command, specifying **reboot** or a future time at the [**edit**] hierarchy level:

```
[edit]
user@host # commit at string
```

Where ***string*** is **reboot** or the future time to activate the configuration changes. You can specify time in two formats:

- A time value in the form ***hh:mm***[***:ss***] hours, minutes, and optionally seconds)—Commit the configuration at the specified time, which must be in the future but before 11:59:59 PM on the day the **commit at** configuration mode command is issued. Use 24-hour time for the ***hh*** value; for example, **04:30:00** is 4:30:00 AM, and **20:00** is 8:00 PM. The time is interpreted with respect to the clock and time zone settings on the router.
- A date and time value in the form ***yyyy-mm-dd hh:mm***[***:ss***] (year, month, date, hours, minutes, and, optionally, seconds)—Commit the configuration at the specified day and time, which must be after the **commit at** command is issued. Use 24-hour time for the ***hh*** value. For example, **2003-08-21 12:30:00** is 12:30 PM on August 21, 2003. The time is interpreted with respect to the clock and time zone settings on the router.

Copyright © 2014, Juniper Networks, Inc.

Enclose the ***string*** value in quotation marks (" "). For example, **commit at "18:00:00"**. For date and time, include both values in the same set of quotation marks. For example, **commit at "2005-03-10 14:00:00".**

A commit check is performed immediately when you issue the **commit at** configuration mode command. If the result of the check is successful, then the current user is logged out of configuration mode, and the configuration data is left in a read-only state. No other commit can be performed until the scheduled commit is completed.

NOTE: If Junos OS fails before the configuration changes become active, all configuration changes are lost.

You cannot enter the **commit at** configuration command after you issue the **request system reboot** command.

You cannot enter the **request system reboot** command once you schedule a commit operation for a specific time in the future.

You cannot commit a configuration when a scheduled commit is pending. For information about how to cancel a scheduled configuration by means of the **clear** command, see the CLI Explorer.

NOTE: We do not recommend performing a commit operation on the backup Routing Engine when graceful Routing Engine switchover is enabled on the router.

Related Documentation

- Committing a Junos OS Configuration on page 98
- Monitoring the Junos OS Commit Process on page 104

Monitoring the Junos OS Commit Process

To monitor the Junos commit process, use the **display detail** command after the pipe with the **commit** command:

```
user@host# commit | display detail
```

For example:

```
[edit]
user@host# commit | display detail
2003-09-22 15:39:39 PDT: exporting juniper.conf
2003-09-22 15:39:39 PDT: setup foreign files
2003-09-22 15:39:39 PDT: propagating foreign files
2003-09-22 15:39:39 PDT: complete foreign files
2003-09-22 15:39:40 PDT: copying configuration to juniper.data+
2003-09-22 15:39:40 PDT: dropping unchanged foreign files
2003-09-22 15:39:40 PDT: daemons checking new configuration
2003-09-22 15:39:41 PDT: commit wrapup...
```


Copyright © 2014, Juniper Networks, Inc.

```
2003-09-22 15:39:42 PDT: activating '/var/etc/ntp.conf'
2003-09-22 15:39:42 PDT: activating '/var/etc/kmd.conf'
2003-09-22 15:39:42 PDT: activating '/var/db/juniper.data'
2003-09-22 15:39:42 PDT: notifying daemons of new configuration
2003-09-22 15:39:42 PDT: signaling 'Firewall daemon', pid 24567, signal 1,
status 0
2003-09-22 15:39:42 PDT: signaling 'Interface daemon', pid 24568, signal 1,
status 0
2003-09-22 15:39:43 PDT: signaling 'Routing protocol daemon', pid 25679,
signal 1, status 0
2003-09-22 15:39:43 PDT: signaling 'MIB2 daemon', pid 24549, signal 1,
status 0
2003-09-22 15:39:43 PDT: signaling 'NTP daemon', pid 37863, signal 1, status 0
2003-09-22 15:39:43 PDT: signaling 'Sonet APS daemon', pid 24551, signal 1,
status 0
2003-09-22 15:39:43 PDT: signaling 'VRRP daemon', pid 24552, signal 1,
status 0
2003-09-22 15:39:43 PDT: signaling 'PFE daemon', pid 2316, signal 1, status 0
2003-09-22 15:39:43 PDT: signaling 'Traffic sampling control daemon', pid 24553
signal 1, status 0
2003-09-22 15:39:43 PDT: signaling 'IPsec Key Management daemon', pid
24556, signal 1, status 0
2003-09-22 15:39:43 PDT: signaling 'Forwarding UDP daemon', pid 2320,
signal 1, status 0
commit complete
```

Related Documentation

- Committing a Junos OS Configuration on page 98
- Adding a Comment to Describe the Committed Configuration on page 105

Adding a Comment to Describe the Committed Configuration

You can include a comment that describes changes to the committed configuration. To do so, include the commit **comment** statement. The comment can be as long as 512 bytes and you must type it on a single line.

```
[edit]
user@host# commit comment comment-string
```

comment-string is the text of the comment.

NOTE: You cannot include a comment with the **commit check** command.

To add a comment to the **commit** command, include the **comment** statement after the **commit** command:

```
[edit]
user@host# commit comment "add user joe"
commit complete
[edit]
user@host#
```

Copyright © 2014, Juniper Networks, Inc.

To add a comment to the **commit confirmed** command, include the **comment** statement after the **commit confirmed** command:

```
[edit]
user@host# commit confirmed comment "add customer to port 27"
commit confirmed will be automatically rolled back in 10 minutes unless confirmed
commit complete
[edit]
user@host#
```

To view these commit comments, issue the **show system commit** operational mode command.

In Junos OS Release 11.4 and later, you can also use the **commit confirmed** command in the **[edit private]** configuration mode.

Related Documentation

- Committing a Junos OS Configuration on page 98
- Backing Up the Committed Configuration on the Alternate Boot Drive on page 106

Backing Up the Committed Configuration on the Alternate Boot Drive

After you commit the configuration and are satisfied that it is running successfully, you should issue the **request system snapshot** command to back up the new software onto the **/altconfig** file system. If you do not issue the **request system snapshot** command, the configuration on the alternate boot drive will be out of sync with the configuration on the primary boot drive.

The **request system snapshot** command backs up the root file system to **/altroot**, and **/config** to **/altconfig**. The root and **/config** file systems are on the router's flash drive, and the **/altroot** and **/altconfig** file systems are on the router's hard disk (if available).

NOTE: To back up the file system on a J Series Services Router, you must specify a media type (primary compact flash drive, removable compact flash drive, or USB storage device) for backup. For more information about the **request system snapshot** command, see the CLI Explorer.

NOTE: For more information about backing up the file system on an ACX Series Universal Access Router, see *Understanding System Snapshot on an ACX Series Router*.

After you issue the **request system snapshot** command, you cannot return to the previous version of the software because the running and backup copies of the software are identical.

Related Documentation

- Committing a Junos OS Configuration on page 98

Copyright © 2014, Juniper Networks, Inc.

Junos OS Batch Commits Overview

Junos OS provides a batch commit feature that aggregates or merges multiple configuration edits from different CLI sessions or users and adds them to a batch commit queue. A batch commit server running on the device takes one or more jobs from the batch commit queue, applies the configuration changes to the shared configuration database, and then commits the configuration changes in a single commit operation.

Batches are prioritized by the commit server based on priority of the batch specified by the user or the time when the batch job is added. When one batch commit is complete, the next set of configuration changes are aggregated and loaded into the batch queue for the next session of the batch commit operation. Batches are created until there are no commit entries left in the queue directory.

When compared to the regular commit operation where all commits are independently committed sequentially, batch commits save time and system resources by committing multiple small configuration edits in a single commit operation.

Batch commits are performed from the **[edit batch]** configuration mode. The commit server properties can be configured at the **[edit system commit server]** hierarchy level.

Aggregation and Error Handling

When there is a load-time error in one of the aggregated jobs, the commit job that encounters the error is discarded and the remaining jobs are aggregated and committed.

For example, if there are five commit jobs (**commit-1**, **commit-2**, **commit-3**, **commit-4**, and **commit-5**) being aggregated, and **commit-3** encounters an error while loading, **commit-3** is discarded and **commit-1**, **commit-2**, **commit-4**, and **commit-5** are aggregated and committed.

If there is an error during the commit operation when two or more jobs are aggregated and committed, the aggregation is discarded and each of those jobs is committed individually like a regular commit operation.

For example, if there are five commit jobs (**commit-1**, **commit-2**, **commit-3**, **commit-4**, and **commit-5**) that are aggregated and if there is a commit error caused because of **commit-3**, the aggregation is discarded, **commit-1**, **commit-2**, **commit-3**, **commit-4**, and **commit-5** are committed individually, and the CLI reports a commit error for **commit-3**.

Related Documentation

- Example: Configuring Batch Commit Server Properties on page 107

Example: Configuring Batch Commit Server Properties

This example shows how to configure batch commit server properties to manage batch commit operations.

- Requirements on page 108
- Overview on page 108

Copyright © 2014, Juniper Networks, Inc.

- Configuration on page 108
- Verification on page 110

Requirements

This example uses the following hardware and software components:

- MX Series 3D Universal Edge Router
- Junos OS Release 12.1 or later running on the device

Overview

You can control how the batch commit queue is handled by the commit server by configuring the server properties at the **[edit system commit server]** hierarchy level. This enables you to control how many commit jobs are aggregated or merged into a single batch commit, the maximum number of jobs that can be added to the queue, days to keep batch commit error logs, interval between two batch commits, and tracing operations for batch commit operations.

Configuration

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level. You can configure the commit server properties from either the regular **[edit]** mode or the **[edit batch]** mode.

Device R0

```
set system commit server maximum-aggregate-pool 4
set system commit server maximum-entries 500
set system commit server commit-interval 5
set system commit server days-to-keep-error-logs 30
set system commit server traceoptions commitd_nov
set system commit server traceoptions flag all
```

Configuring the Commit Server Properties

Step-by-Step Procedure

1. (Optional) Configure the number of commit transactions to aggregate or merge in a single commit operation.

   The default value for **maximum-aggregate-pool** is **5**.

   NOTE: **Setting maximum-aggregate-pool to 1 commits each of the jobs individually.**

   In this example, the number of commit transactions is set to **4** indicating that four different commit jobs are aggregated into a single commit before the commit operation is initiated.

   ```
   [edit system commit server]
   user@R0# set maximum-aggregate-pool 4
   ```


Copyright © 2014, Juniper Networks, Inc.

2. (Optional) Configure the maximum number of jobs allowed in a batch.

   This limits the number of commits jobs that are added to the queue.

   ```
   [edit system commit server]
   user@R0# set maximum-entries 500
   ```

   NOTE: If you set **maximum-entries** to 1, the commit server cannot add more than one job to the queue, and the CLI displays an appropriate message when you try to commit more than one job.

3. (Optional) Configure the time (in seconds) to wait before starting the next batch commit operation.

   ```
   [edit system commit server]
   user@R0# set commit-interval 5
   ```

4. (Optional) Configure the number of days to keep error logs.

   The default value is **30** days.

   ```
   [edit system commit server]
   user@R0# set days-to-keep-error-logs 30
   ```

5. (Optional) Configure tracing operations to log batch commit events.

   In this example, the filename for logging batch commit events is **commitd_nov**, and all traceoption flags are set.

   ```
   [edit system commit server]
   user@R0# set traceoptions commitd_nov
   user@R0# set traceoptions flag all
   ```

Results From configuration mode, confirm your configuration by entering the **show system commit server** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R0# show system commit server
maximum-aggregate-pool 4;
maximum-entries 500;
commit-interval 5;
days-to-keep-error-logs 30;
traceoptions {
   file commitd_nov;
   flag all;
}
```

Committing the Configuration from Batch Configuration Mode

Step-by-Step Procedure To commit the configuration from the **[edit batch]** mode, do one of the following:

- Log in to the device and enter **commit**.

  ```
  [edit batch]
  user@R0# commit
  ```

Copyright © 2014, Juniper Networks, Inc.

```
Added to commit queue request-id: 1000
```

- To assign a higher priority to a batch commit job, issue the **commit** command with the **priority** option.

```
[edit batch]
user@R0# commit priority
  Added to commit queue request-id: 1001
```

- To commit a configuration without aggregating the configuration changes with other commit jobs in the queue, issue the **commit** command with the **atomic** option.

```
[edit batch]
user@R0# commit atomic
  Added to commit queue request-id: 1002
```

- To commit a configuration without aggregating the configuration changes with other commit jobs in the queue, and issuing a higher priority to the commit job, issue the **commit** command with the **atomic priority** option.

```
[edit batch]
user@R0# commit atomic priority
  Added to commit queue request-id: 1003
```

Verification

Confirm that the configuration is working properly.

- Checking the Batch Commit Server Status on page 110
- Checking the Batch Commit Status on page 111
- Viewing the Patch Files in a Batch Commit Job on page 111
- Viewing the Trace Files for Batch Commit Operations on page 113

Checking the Batch Commit Server Status

Purpose Check the status of the batch commit server.

Action

```
user@R0> show system commit server
Commit server status : Not running
```

By default, the status of the commit server is **Not running**. The commit server starts running only when a batch commit job is added to the queue.

When a batch commit job is added to the queue, the status of the commit server changes to **Running**.

```
user@R0> show system commit server

Commit server status : Running
Jobs in process:
  1003 1004 1005
```

Meaning The **Jobs in process** field lists the commit IDs of jobs that are in process.

Copyright © 2014, Juniper Networks, Inc.

Checking the Batch Commit Status

Purpose Check the commit server queue for the status of the batch commits.

Action

```
user@R0> show system commit server queue

Pending commits:
  Id: 1005
  Last Modified: Tue Nov  1 23:56:43 2011

Completed commits:
  Id: 1000
  Last Modified: Tue Nov  1 22:46:43 2011
  Status: Successfully committed 1000

  Id: 1002
  Last Modified: Tue Nov  1 22:50:35 2011
  Status: Successfully committed 1002

  Id: 1004
  Last Modified: Tue Nov  1 22:51:48 2011
  Status: Successfully committed 1004

  Id: 1007
  Last Modified: Wed Nov  2 01:08:04 2011
  Status: Successfully committed 1007

  Id: 1009
  Last Modified: Wed Nov  2 01:16:45 2011
  Status: Successfully committed 1009

  Id: 1010
  Last Modified: Wed Nov  2 01:19:25 2011
  Status: Successfully committed 1010

  Id: 1011
  Last Modified: Wed Nov  2 01:28:16 2011
  Status: Successfully committed 1011


Error commits:
  Id: 1008
  Last Modified: Wed Nov  2 01:08:18 2011
  Status: Error while commiting 1008
```

Meaning **Pending commits** displays commit jobs that are added to the commit queue but are not committed yet. **Completed commits** displays the list of commit jobs that are successful. **Error commits** are commits that failed because of an error.

Viewing the Patch Files in a Batch Commit Job

Purpose View the timestamps, patch files, and the status of each of the commit jobs. Patch files show the configuration changes that occur in each commit operation that is added to the batch commit queue.

Copyright © 2014, Juniper Networks, Inc.

Action 1. Issue the **show system commit server queue patch** command to view the patches for all commit operations.

```
user@R0> show system commit server queue patch
Pending commits:
  none


Completed commits:
  Id: 1000
  Last Modified: Tue Nov  1 22:46:43 2011
  Status: Successfully committed 1000

Patch:
[edit groups]
  re1 { ... }
+ GRP-DHCP-POOL-NOACCESS {
+     access {
+         address-assignment {
+             pool <*> {
+                 family inet {
+                     dhcp-attributes {
+                         maximum-lease-time 300;
+                         grace-period 300;
+                         domain-name verizon.net;
+                         name-server {
+                             4.4.4.1;
+                             4.4.4.2;
+                         }
+                     }
+                 }
+             }
+         }
+     }
+ }
  Id: 1002
  Last Modified: Tue Nov  1 22:50:35 2011
  Status: Successfully committed 1002

Patch:
[edit]
+ snmp {
+     community abc;
+ }
  Id: 1010
  Last Modified: Wed Nov  2 01:19:25 2011
  Status: Successfully committed 1010

Patch:
[edit system syslog]
  file test { ... }
+ file j {
+     any any;
+ }

Error commits:
  Id: 1008
  Last Modified: Wed Nov  2 01:08:18 2011
  Status: Error while commiting 1008

Patch:
```


Copyright © 2014, Juniper Networks, Inc.

```
[edit system]
+ radius-server {
+     10.1.1.1 port 222;
+ }
```

The output shows the changes in configuration for each commit job ID.

2. To view the patch for a specific commit job ID, issue the **show system commit server queue patch id *<id-number>*** command.

```
user@R0> show system commit server queue patch id 1000
Completed commits:
  Id: 1000
  Last Modified: Tue Nov  1 22:46:43 2011
  Status: Successfully committed 1000

Patch:
[edit system]
+ radius-server {
+     192.168.69.162 secret teH.bTc/RVbPM;
+     192.168.64.10 secret teH.bTc/RVbPM;
+     192.168.60.52 secret teH.bTc/RVbPM;
+     192.168.60.55 secret teH.bTc/RVbPM;
+     192.168.4.240 secret teH.bTc/RVbPM;
+ }
```

Meaning The output shows the patch created for a commit job. The **+** or **-** sign indicates the changes in the configuration for a specific commit job.

Viewing the Trace Files for Batch Commit Operations

Purpose View the trace files for batch commit operations. You can use the trace files for troubleshooting purposes.

Copyright © 2014, Juniper Networks, Inc.

Action

- Issue the **file show /var/log/*<filename>*** command to view all entries in the log file.

  user@R0> **file show/var/log/commitd_nov**

  The output shows commit server event logs and other logs for batch commits.

```
Nov  1 22:46:43 Successfully committed 1000
Nov  1 22:46:43 pausing after commit for 0 seconds
...
Nov  1 22:46:43 Done working on queue
...

Nov  1 22:47:17 maximum-aggregate-pool = 5
Nov  1 22:47:17 maximum-entries= 0
Nov  1 22:47:17 asynchronous-prompt = no
Nov  1 22:47:17 commit-interval = 0
Nov  1 22:47:17 days-to-keep-error-logs = -1
...
Nov  1 22:47:17 Added to commit queue request-id: 1001
Nov  1 22:47:17 Commit server status=running
Nov  1 22:47:17 No need to pause
...

Nov  1 22:47:18 Error while commiting 1001
Nov  1 22:47:18 doing rollback
...
```

- To view log entries only for successful batch commit operations, issue the **file show /var/log/*<filename>*** command with the **| match committed** pipe option.

  user@R0> **file show/var/log/commitd_nov | match committed**

  The output shows batch commit job IDs for successful commit operations.

```
Nov  1 22:46:43 Successfully committed 1000
Nov  1 22:50:35 Successfully committed 1002
Nov  1 22:51:48 Successfully committed 1004
Nov  2 01:08:04 Successfully committed 1007
Nov  2 01:16:45 Successfully committed 1009
Nov  2 01:19:25 Successfully committed 1010
Nov  2 01:28:16 Successfully committed 1011
```

- To view log entries only for failed batch commit operations, issue the **file show /var/log/*<filename>*** command with the **| match “Error while”** pipe option.

  user@R0> **file show/var/log/commitd_nov | match “Error while”**

  The output shows commit job IDs for failed commit operations.

```
Nov  1 22:47:18 Error while commiting 1001
Nov  1 22:51:10 Error while commiting 1003
Nov  1 22:52:15 Error while commiting 1005
...
```

- To view log entries only for commit server events, issue the **file show /var/log/*<filename>*** command with the **| match “commit server”** pipe option.

  user@R0> **file show/var/log/commitd_nov | match “commit server”**

  The output shows commit server event logs.

```
Nov  1 22:46:39 Commit server status=running
Nov  1 22:46:39 Commit server jobs=1000
```


Copyright © 2014, Juniper Networks, Inc.

```
Nov  1 22:46:43 Commit server status=not running
Nov  1 22:46:43 Commit server jobs=
Nov  1 22:47:17 Commit server status=running
Nov  1 22:47:18 Commit server jobs=1001
Nov  1 22:47:18 2 errors reported by commit server
Nov  1 22:47:18 Commit server status=not running
Nov  1 22:47:18 Commit server jobs=
Nov  1 22:50:31 Commit server status=running
Nov  1 22:50:31 Commit server jobs=1002
Nov  1 22:50:35 Commit server status=not running
Nov  1 22:50:35 Commit server jobs=
Nov  1 22:51:09 Commit server status=running
Nov  1 22:51:10 Commit server jobs=1003
Nov  1 22:51:10 2 errors reported by commit server
Nov  1 22:51:10 Commit server status=not running
...
```

Related Documentation

- Junos OS Batch Commits Overview on page 107
- commit-interval (Batch Commits) on page 245
- days-to-keep-error-logs (Batch Commits) on page 246
- maximum-aggregate-pool (Batch Commits) on page 256
- maximum-entries (Batch Commits) on page 257
- maximum-entries on page 257
- server (Batch Commits) on page 265
- traceoptions (Batch Commits) on page 278

Copyright © 2014, Juniper Networks, Inc.

Copyright © 2014, Juniper Networks, Inc.

CHAPTER 6

Managing Configurations

- Understanding How the Junos OS Configuration Is Stored on page 117
- Comparing Configuration Changes with a Prior Version on page 118
- Returning to the Most Recently Committed Junos OS Configuration on page 120
- Returning to a Previously Committed Junos OS Configuration on page 120
- Saving a Configuration to a File on page 125
- Additional Details About Specifying Junos OS Statements and Identifiers on page 126
- Loading a Configuration from a File on page 129
- Examples: Loading a Configuration from a File on page 132
- Creating and Returning to a Rescue Configuration on page 135
- Compressing the Current Configuration File on page 135
- Example: Protecting the Junos OS Configuration from Modification or Deletion on page 137
- Synchronizing Routing Engines on page 144
- Configuring Multiple Routing Engines to Synchronize Committed Configurations Automatically on page 147

Understanding How the Junos OS Configuration Is Stored

When you edit a configuration, you work in a copy of the current configuration to create a candidate configuration. The changes you make to the candidate configuration are visible in the CLI immediately, so if multiple users are editing the configuration at the same time, all users can see all changes.

To have a candidate configuration take effect, you *commit* the changes. At this point, the candidate file is checked for proper syntax, activated, and marked as the current, operational software configuration file. If multiple users are editing the configuration, when you commit the candidate configuration, all changes made by all the users take effect.

In addition to saving the current configuration, the CLI saves the current operational version and the previous 49 versions of committed configurations. The most recently committed configuration is version 0, which is the current operational version and the

Copyright © 2014, Juniper Networks, Inc.

default configuration that the system returns to if you roll back to a previous configuration. The oldest saved configuration is version 49.

By default, the Junos OS saves the current configuration and three previous versions of the committed configuration on the CompactFlash card. The currently operational Junos OS configuration is stored in the file juniper.conf.gz, and the last three committed configurations are stored in the files juniper.conf.1.gz, juniper.conf.2.gz, and juniper.conf.3.gz. These four files are located in the router or switch's CompactFlash card in the directory /config.

The remaining 46 previous versions of committed configurations, the files **juniper.conf.4** through **juniper.conf.49**, are stored in the directory **/var/db/config** on the hard disk.

Related Documentation

- *Using Junos OS to Specify the Number of Configurations Stored on the CompactFlash Card*
- Returning to the Most Recently Committed Junos OS Configuration on page 120
- Returning to a Previously Committed Junos OS Configuration on page 120
- Loading a Configuration from a File on page 129

Comparing Configuration Changes with a Prior Version

In configuration mode only, when you have made changes to the configuration and want to compare the candidate configuration with a prior version, you can use the **compare** command to display the configuration. The **compare** command compares the candidate configuration with either the current committed configuration or a configuration file and displays the differences between the two configurations. To compare configurations, specify the **compare** command after the pipe:

```
[edit]
user@host# show | compare (filename| rollback n)
```

filename is the full path to a configuration file. The file must be in the proper format: a hierarchy of statements.

n is the index into the list of previously committed configurations. The most recently saved configuration is number 0, and the oldest saved configuration is number 49. If you do not specify arguments, the candidate configuration is compared against the active configuration file (**/config/juniper.conf**).

The comparison output uses the following conventions:

- Statements that are only in the candidate configuration are prefixed with a plus sign (+).
- Statements that are only in the comparison file are prefixed with a minus sign (-).
- Statements that are unchanged are prefixed with a single blank space ( ).

Copyright © 2014, Juniper Networks, Inc.

The following example shows various changes, then a comparison of the candidate configuration with the active configuration, showing only the changes made at the **[edit protocols bgp]** hierarchy level:

```
[edit]
user@host# edit protocols bgp
[edit protocols bgp]
user@host# show
group my-group {
    type internal;
    hold-time 60;
    advertise-inactive;
    allow 1.1.1.1/32;
}
group fred {
    type external;
    peer-as 33333;
    allow 2.2.2.2/32;
}
group test-peers {
    type external;
    allow 3.3.3.3/32;
}
[edit protocols bgp]
user@host# set group my-group hold-time 90
[edit protocols bgp]
user@host# delete group my-group advertise-inactive
[edit protocols bgp]
user@host# set group fred advertise-inactive
[edit protocols bgp]
user@host# delete group test-peers
[edit protocols bgp]
user@host# show | compare
[edit protocols bgp group my-group]
-hold-time 60;
+hold-time 90;
-advertise-inactive;
[edit protocols bgp group fred]
+advertise-inactive;
[edit protocols bgp]
-group test-peers {
    -type external;
    -allow 3.3.3.3/32;
}
[edit protocols bgp]
user@host# show
group my-group {
    type internal;
    hold-time 90;
    allow 1.1.1.1/32;
}
group fred {
    type external;
    advertise-inactive;
    peer-as 3333;
```

Copyright © 2014, Juniper Networks, Inc.

```
    allow 2.2.2.2/32;
}
```

Related Documentation

- Creating and Returning to a Rescue Configuration on page 123

Returning to the Most Recently Committed Junos OS Configuration

To return to the most recently committed configuration and load it into configuration mode without activating it, use the **rollback** configuration mode command:

```
[edit]
user@host# rollback
load complete
```

To activate the configuration to which you rolled back, use the **commit** command:

```
[edit]
user@host# rollback
load complete
[edit]
user@host# commit
```

Related Documentation

- Rolling Back Junos OS Configuration Changes on page 46
- Returning to a Previously Committed Junos OS Configuration on page 120
- Understanding How the Junos OS Configuration Is Stored on page 117

Returning to a Previously Committed Junos OS Configuration

This topic explains how you can return to a configuration prior to the most recently committed one, and contains the following sections:

- Returning to a Configuration Prior to the One Most Recently Committed on page 120
- Displaying Previous Configurations on page 121
- Comparing Configuration Changes with a Prior Version on page 122
- Creating and Returning to a Rescue Configuration on page 123
- Saving a Configuration to a File on page 124

Returning to a Configuration Prior to the One Most Recently Committed

To return to a configuration prior to the most recently committed one, include the configuration number, 0 through 49, in the **rollback** command. The most recently saved configuration is number 0 (which is the default configuration to which the system returns), and the oldest saved configuration is number 49.

```
[edit]
user@host# rollback number
load complete
```


Copyright © 2014, Juniper Networks, Inc.

Displaying Previous Configurations

To display previous configurations, including the rollback number, date, time, the name of the user who committed changes, and the method of commit, use the **rollback ?** command.

```
[edit]
user@host# rollback ?
Possible completions:
<[Enter]> Execute this command
<number> Numeric argument
0              2005-02-27 12:52:10 PST by abc via cli
1              2005-02-26 14:47:42 PST by def via cli
2              2005-02-14 21:55:45 PST by ghi via cli
3              2005-02-10 16:11:30 PST by jkl via cli
4              2005-02-10 16:02:35 PST by mno via cli
5              2005-03-16 15:10:41 PST by pqr via cli
6              2005-03-16 14:54:21 PST by stu via cli
7              2005-03-16 14:51:38 PST by vwx via cli
8              2005-03-16 14:43:29 PST by yzz via cli
9              2005-03-16 14:15:37 PST by abc via cli
10             2005-03-16 14:13:57 PST by def via cli
11             2005-03-16 12:57:19 PST by root via other
12             2005-03-16 10:45:23 PST by root via other
13             2005-03-16 10:08:13 PST by root via other
14             2005-03-16 01:20:56 PST by root via other
15             2005-03-16 00:40:37 PST by ghi via cli
16             2005-03-16 00:39:29 PST by jkl via cli
17             2005-03-16 00:32:36 PST by mno via cli
18             2005-03-16 00:31:17 PST by pqr via cli
19             2005-03-15 19:59:00 PST by stu via cli
20             2005-03-15 19:53:39 PST by vwx via cli
21             2005-03-15 18:07:19 PST by yzz via cli
22             2005-03-15 17:59:03 PST by abc via cli
23             2005-03-15 15:05:14 PST by def via cli
24             2005-03-15 15:04:51 PST by ghi via cli
25             2005-03-15 15:03:42 PST by jkl via cli
26             2005-03-15 15:01:52 PST by mno via cli
27             2005-03-15 14:58:34 PST by pqr via cli
28             2005-03-15 13:09:37 PST by root via other
29             2005-03-12 11:01:20 PST by stu via cli
30             2005-03-12 10:57:35 PST by vwx via cli
31             2005-03-11 10:25:07 PST by yzz via cli
32             2005-03-10 23:40:58 PST by abc via cli
33             2005-03-10 23:40:38 PST by def via cli
34             2005-03-10 23:14:27 PST by ghi via cli
35             2005-03-10 23:10:16 PST by jkl via cli
36             2005-03-10 23:01:51 PST by mno via cli
37             2005-03-10 22:49:57 PST by pqr via cli
38             2005-03-10 22:24:07 PST by stu via cli
39             2005-03-10 22:20:14 PST by vwx via cli
40             2005-03-10 22:16:56 PST by yzz via cli
41             2005-03-10 22:16:41 PST by abc via cli
42             2005-03-10 20:44:00 PST by def via cli
43             2005-03-10 20:43:29 PST by ghi via cli
```

Copyright © 2014, Juniper Networks, Inc.

```
44              2005-03-10 20:39:14 PST by jkl via cli
45              2005-03-10 20:31:30 PST by root via other
46              2005-03-10 18:57:01 PST by mno via cli
47              2005-03-10 18:56:18 PST by pqr via cli
48              2005-03-10 18:47:49 PST by stu via cli
49              2005-03-10 18:47:34 PST by vw via cli
|Pipe through a command
[edit]
```

Comparing Configuration Changes with a Prior Version

In configuration mode only, when you have made changes to the configuration and want to compare the candidate configuration with a prior version, you can use the **compare** command to display the configuration. The **compare** command compares the candidate configuration with either the current committed configuration or a configuration file and displays the differences between the two configurations. To compare configurations, specify the **compare** command after the pipe:

```
[edit]
user@host# show | compare (filename| rollback n)
```

filename is the full path to a configuration file. The file must be in the proper format: a hierarchy of statements.

n is the index into the list of previously committed configurations. The most recently saved configuration is number 0, and the oldest saved configuration is number 49. If you do not specify arguments, the candidate configuration is compared against the active configuration file (**/config/juniper.conf**).

The comparison output uses the following conventions:

- Statements that are only in the candidate configuration are prefixed with a plus sign (+).
- Statements that are only in the comparison file are prefixed with a minus sign (-).
- Statements that are unchanged are prefixed with a single blank space ( ).

The following example shows various changes, then a comparison of the candidate configuration with the active configuration, showing only the changes made at the **[edit protocols bgp]** hierarchy level:

```
[edit]
user@host# edit protocols bgp
[edit protocols bgp]
user@host# show
group my-group {
    type internal;
    hold-time 60;
    advertise-inactive;
    allow 1.1.1.1/32;
}
group fred {
    type external;
    peer-as 33333;
```


Copyright © 2014, Juniper Networks, Inc.

```
    allow 2.2.2.2/32;
}
group test-peers {
    type external;
    allow 3.3.3.3/32;
}
[edit protocols bgp]
user@host# set group my-group hold-time 90
[edit protocols bgp]
user@host# delete group my-group advertise-inactive
[edit protocols bgp]
user@host# set group fred advertise-inactive
[edit protocols bgp]
user@host# delete group test-peers
[edit protocols bgp]
user@host# show | compare
[edit protocols bgp group my-group]
-hold-time 60;
+hold-time 90;
-advertise-inactive;
[edit protocols bgp group fred]
+advertise-inactive;
[edit protocols bgp]
-group test-peers {
    -type external;
    -allow 3.3.3.3/32;
}
[edit protocols bgp]
user@host# show
group my-group {
    type internal;
    hold-time 90;
    allow 1.1.1.1/32;
}
group fred {
    type external;
    advertise-inactive;
    peer-as 3333;
    allow 2.2.2.2/32;
}
```

Creating and Returning to a Rescue Configuration

A *rescue configuration* allows you to define a known working configuration or a configuration with a known state that you can roll back to at any time. This alleviates the necessity of having to remember the rollback number with the **rollback** command. You use the rescue configuration when you need to roll back to a known configuration or as a last resort if your router or switch configuration and the backup configuration files become damaged beyond repair.

To save the most recently committed configuration as the rescue configuration so that you can return to it at any time, issue the **request system configuration rescue save** command:

Copyright © 2014, Juniper Networks, Inc.

```
user@host> request system configuration rescue save
```

To return to the rescue configuration, use the **rollback rescue** configuration mode command:

```
[edit]
user@host# rollback rescue
load complete
```

NOTE: If the rescue configuration does not exist, or if the rescue configuration is not a complete, viable configuration, the rollback command fails, an error message appears, and the current configuration remains active.

To activate the rescue configuration that you have loaded, use the **commit** command:

```
[edit]
user@host# rollback rescue
load complete
[edit]
user@host# commit
```

To delete an existing rescue configuration, issue the **request system configuration rescue delete** command:

```
user@host> request system configuration rescue delete
user@host>
```

For more information about the **request system configuration rescue delete** and **request system configuration rescue save** commands, see the CLI Explorer.

Saving a Configuration to a File

Save the Junos OS configuration to a file so that you can edit it with a text editor of your choice. You can save your current configuration to an ASCII file, which saves the configuration in its current form, including any uncommitted changes. If more than one user is modifying the configuration, all changes made by all users are saved.

To save software configuration changes to an ASCII file, use the **save** configuration mode command:

```
[edit]
user@host# save filename
[edit]
user@host#
```

The contents of the current level of the statement hierarchy (and below) are saved, along with the statement hierarchy containing it. This allows a section of the configuration to be saved, while fully specifying the statement hierarchy.

By default, the configuration is saved to a file in your home directory, which is on the flash drive.

Copyright © 2014, Juniper Networks, Inc.

When you issue this command from anywhere in the hierarchy (except the top level), a **replace** tag is automatically included at the beginning of the file. You can use the **replace** tag to control how a configuration is loaded from a file.

```
user@host> file show /var/home/user/myconf
replace:
protocols {
    bgp {
        disable;
        group int {
            type internal;
        }
    }
    isis {
        disable;
        interface all {
            level 1 disable;
        }
        interface fxp0.0 {
            disable;
        }
    }
    ospf {
        traffic-engineering;
        reference-bandwidth 4g;
        ...
    }
}
```

Related Documentation

- Returning to the Most Recently Committed Junos OS Configuration on page 120
- Loading a Configuration from a File on page 129
- Specifying Filenames and URLs on page 162

Saving a Configuration to a File

Save the Junos OS configuration to a file so that you can edit it with a text editor of your choice. You can save your current configuration to an ASCII file, which saves the configuration in its current form, including any uncommitted changes. If more than one user is modifying the configuration, all changes made by all users are saved.

To save software configuration changes to an ASCII file, use the **save** configuration mode command:

```
[edit]
user@host# save filename
[edit]
user@host#
```

The contents of the current level of the statement hierarchy (and below) are saved, along with the statement hierarchy containing it. This allows a section of the configuration to be saved, while fully specifying the statement hierarchy.

Copyright © 2014, Juniper Networks, Inc.

By default, the configuration is saved to a file in your home directory, which is on the flash drive.

When you issue this command from anywhere in the hierarchy (except the top level), a **replace** tag is automatically included at the beginning of the file. You can use the **replace** tag to control how a configuration is loaded from a file.

```
user@host> file show /var/home/user/myconf
replace:
protocols {
    bgp {
        disable;
        group int {
            type internal;
        }
    }
    isis {
        disable;
        interface all {
            level 1 disable;
        }
        interface fxp0.0 {
            disable;
        }
    }
    ospf {
        traffic-engineering;
        reference-bandwidth 4g;
        ...
    }
}
```

Additional Details About Specifying Junos OS Statements and Identifiers

This topic provides more detailed information about CLI container and leaf statements so that you can better understand how you must specify them when creating ASCII configuration files. It also describes how the CLI performs type checking to verify that the data you entered is in the correct format.

- Specifying Statements on page 126
- Performing CLI Type Checking on page 128

Specifying Statements

Statements are shown one of two ways, either with braces or without:

- Statement name and identifier, with one or more lower level statements enclosed in braces:

  ```
  statement-name1 identifier-name {
      statement-name2;
      additional-statements;
  }
  ```

- Statement name, identifier, and a single identifier:

Copyright © 2014, Juniper Networks, Inc.

statement-name identifier-name1 identifier-name2;

The ***statement-name*** is the name of the statement.

The ***identifier-name*** is a name or other string that uniquely identifies an instance of a statement. An identifier is used when a statement can be specified more than once in a configuration.

When specifying a statement, you must specify either a statement name or an identifier name, or both, depending on the statement hierarchy.

You specify identifiers in one of the following ways:

- ***identifier-name***—The ***identifier-name*** is a keyword used to uniquely identify a statement when a statement can be specified more than once in a statement.
- ***identifier-name value***—The ***identifier-name*** is a keyword, and the ***value*** is a required option variable.
- ***identifier-name* [*value1 value2 value3 ...*]**—The ***identifier-name*** is a keyword that accepts multiple values. The brackets are required when you specify a set of values; however, they are optional when you specify only one value.

The following examples illustrate how statements and identifiers are specified in the configuration:

```
protocol {      # Top-level statement (statement-name).
 ospf {          # Statement under "protocol" (statement-name).
  area 0.0.0.0 {     # OSPF area "0.0.0.0" (statement-name identifier-name),
   interface so-0/0/0 { # which contains an interface named "so-0/0/0."
    hello-interval 25; # Identifier and value (identifier-name value).
    priority 2;     # Identifier and value (identifier-name value).
    disable;      # Flag identifier (identifier-name).
   }
   interface so-0/0/1;   # Another instance of "interface," named so-0/0/1,
  }             # this instance contains no data, so no braces
 }               # are displayed.
}
policy-options {    # Top-level statement (statement-name).
 term term1 {     # Statement under "policy-options"
           # (statement-name value).
  from {       # Statement under "term" (statement-name).
   route-filter 10.0.0.0/8 orlonger reject;   # One identifier ("route-filter")
with
   route-filter 127.0.0.0/8 orlonger reject;   # multiple values.
   route-filter 128.0.0.0/16 orlonger reject;
   route-filter 149.20.64.0/24 orlonger reject;
   route-filter 172.16.0.0/12 orlonger reject;
   route-filter 191.255.0.0/16 orlonger reject;
  }
  then {       # Statement under "term" (statement-name).
   next term;    # Identifier (identifier-name).
  }
 }
}
```

Copyright © 2014, Juniper Networks, Inc.

When you create an ASCII configuration file, you can specify statements and identifiers in one of the following ways. However, each statement has a preferred style, and the CLI uses that style when displaying the configuration in response to a configuration mode **show** command.

- Statement followed by identifiers:

  statement-name identifier-name [...] ***identifier-name value*** [...];

- Statement followed by identifiers enclosed in braces:

  statement-name {
  identifier-name;
  [...]
  identifier-name value;
  [...]
  }

- For some repeating identifiers, you can use one set of braces for all the statements:

  statement-name {
  identifier-name value1;
  identifier-name value2;
  }

Performing CLI Type Checking

When you specify identifiers and values, the CLI performs type checking to verify that the data you entered is in the correct format. For example, for a statement in which you must specify an IP address, the CLI requires you to enter an address in a valid format. If you have not, an error message indicates what you need to type. Table 7 on page 128 lists the data types the CLI checks.

Table 7: CLI Configuration Input Types

Data Type	Format	Examples
Physical interface name (used in the **[edit interfaces]** hierarchy)	***type-fpc/pic/port***	**Correct:** so-0/0/1 **Incorrect:** so-0
Full interface name	***type-fpc/pic/port<:channel>.logical***	**Correct:** so-0/0/1.0 **Incorrect:** so-0/0/1
Full or abbreviated interface name (used in places other than the **[edit interfaces]** hierarchy)	***type-<fpc</pic/port>><<:channel>.logical>***	**Correct:** so, so-1, so-1/2/3:4.5

Copyright © 2014, Juniper Networks, Inc.

Table 7: CLI Configuration Input Types *(continued)*

Data Type	Format	Examples
IP address	0x*hex-bytesoctet*<.*octet*<.*octet*.<*octet*>>>	**Correct:** 1.2.3.4, 0x01020304, 128.8.1, 128.8 **Sample translations:** **1.2.3** becomes **1.2.3.0** **0x01020304** becomes **1.2.3.4** **0x010203** becomes **0.1.2.3**
IP address (destination prefix) and prefix length	0x*hex-bytes*</*length*>*octet*<*octet*<*octet*.<*octet*>>></*length*>	**Correct:** 10/8, 128.8/16, 1.2.3.4/32, 1.2.3.4 **Sample translations:** **1.2.3** becomes **1.2.3.0/32** **0x01020304** becomes **1.2.3.4/32** **0x010203** becomes **0.1.2.3/32** **default** becomes **0.0.0.0/0**
International Organization for Standardization (ISO) address	*hex-nibble*<*hex-nibble* ...>	**Correct:** 47.1234.2345.3456.00, 47123423453456.00, 47.12.34.23.45.34.56.00 **Sample translations:** **47123456** becomes **47.1234.56** **47.12.34.56** becomes **47.1234.56** **4712.3456** becomes **47.1234.56**
OSPF area identifier (ID)	0x*hex-bytesoctet*<.*octet*<.*octet*.< *octet* >>> *decimal-number*	**Correct:** 54, 0.0.0.54, 0x01020304, 1.2.3.4 **Sample translations:** **54** becomes **0.0.0.54** **257** becomes **0.0.1.1** **128.8** becomes **128.8.0.0** **0x010203** becomes **0.1.2.3**

Related Documentation

- Entering and Exiting the Junos OS CLI Configuration Mode on page 64

Loading a Configuration from a File

You can create a file, copy the file to the local router, and then load the file into the CLI. After you have loaded the file, you can commit it to activate the configuration on the router, or you can edit the configuration interactively using the CLI and commit it at a later time.

You can also create a configuration while typing at the terminal and then load it. Loading a configuration from the terminal is generally useful when you are cutting existing portions of the configuration and pasting them elsewhere in the configuration.

To load an existing configuration file that is located on the router, use the **load** configuration mode command:

Copyright © 2014, Juniper Networks, Inc.

```
[edit]
user@host# load (factory-default | merge | override | patch | replace | set | update)
  filename <relative>
```

For information about specifying the filename, see "Specifying Filenames and URLs" on page 162.

To load a configuration from the terminal, use the following version of the **load** configuration mode command. Press Ctrl-d to end input.

```
[edit]
user@host# load (factory-default | merge | override | patch | replace | set | update)
terminal <relative>
```

To replace an entire configuration, specify the **override** option at any level of the hierarchy. A **load override** operation completely replaces the current candidate configuration with the file you are loading. Thus, if you saved a complete configuration, use this option.

An **override** operation discards the current candidate configuration and loads the configuration in ***filename*** or the configuration that you type at the terminal. When you use the **override** option and commit the configuration, all system processes reparse the configuration. For an example, see Figure 6 on page 132.

To replace portions of a configuration, specify the **replace** option. The **load replace** operation looks for **replace:** tags that you added to the loaded file, and replaces the parts of the candidate configuration with whatever is specified after the tag. This is useful when you want more control over exactly what is being changed. For this operation to work, you must include **replace:** tags in the file or configuration you type at the terminal. The software searches for the **replace:** tags, deletes the existing statements of the same name, if any, and replaces them with the incoming configuration. If there is no existing statement of the same name, the **replace** operation adds to the configuration the statements marked with the **replace:** tag. For an example, see Figure 7 on page 133.

If, in an **override** or **merge** operation, you specify a file or type text that contains **replace:** tags, the **replace:** tags are ignored and the **override** or **merge** operation is performed.

If you are performing a **replace** operation and the file you specify or text you type does not contain any **replace:** tags, the **replace** operation is effectively equivalent to a **merge** operation. This might be useful if you are running automated scripts and cannot know in advance whether the scripts need to perform a **replace** or a **merge** operation. The scripts can use the **replace** operation to cover either case.

The **load merge** operation adds the saved file to the existing candidate configuration. This is useful if you are adding new configuration sections. For example, suppose that you are adding a BGP configuration to the **[edit protocols]** hierarchy level, where there was no BGP configuration before, you can use the **load merge** operation to combine the saved file configuration to the existing candidate configuration. If the existing configuration and the incoming configuration contain conflicting statements, the statements in the incoming configuration override those in the existing configuration.

To replace only the configuration that has changed, specify the **update** option at any level of the hierarchy. The **load update** operation compares the candidate configuration and the file you are loading, and only changes the parts of the candidate configuration

 Copyright © 2014, Juniper Networks, Inc.

that are different from the new configuration. You would use this, for example, if there is an existing BGP configuration and the file you are loading changes it in some way.

To change part of the configuration with a patch file, specify the **patch** option. The **load patch** operation loads a file or terminal input that contains configuration changes. First, on a device that already has the configuration changes, you type the **show | compare** command to output the differences between two configurations. Then you can load the differences on another router. The advantage of the **load patch** command is that it saves you from having to copy snippets from different hierarchy levels into a text file prior to loading them into the target device. This might be a useful time saver if you are configuring several devices with the same options. For example, suppose that you configure a routing policy on Device router1 and you want to replicate the policy configuration on Device router2, router3, and router4, you can use the **load patch** operation.

First, run the **show | compare** command.

```
user@router1# show | compare rollback 3
[edit protocols ospf]
+    export default-static;
-    export static-default
[edit policy-options]
+    policy-statement default-static {
+        from protocol static;
+        then accept;
+    }
```

Copy the output of the **show | compare** command to the clipboard, making sure to include the hierarchy levels. On Device router2, router3, and router4, type **load patch terminal** and paste the output. Press Enter and then press Ctrl-d to end the operation. If the patch input specifies different values for an existing statement, the patch input overrides the existing statement.

To use the **merge**, **replace**, **set**, or **update** option without specifying the full hierarchy level, specify the **relative** option. For example:

```
[edit system]
user@host# show static-host-mapping
bob sysid 987.654.321ab
[edit system]
user@host# load replace terminal relative
[Type ^D at a new line to end input]
replace: static-host-mapping {
    bob sysid 0123.456.789bc;
}
load complete
[edit system]
user@host# show static-host-mapping
bob sysid 0123.456.789bc;
```

To load a configuration that contains the **set** configuration mode command, specify the **set** option. This option executes the configuration instructions line by line as they are stored in a file or from a terminal. The instructions can contain any configuration mode command, such as **set**, **edit**, **exit**, and **top**. For an example, see Figure 10 on page 134.

Copyright © 2014, Juniper Networks, Inc.

To copy a configuration file from another network system to the local router, you can use the SSH and Telnet utilities, as described in the CLI Explorer.

NOTE: If you are using Junos OS in a Common Criteria environment, system log messages are created whenever a secret attribute is changed (for example, password changes or changes to the RADIUS shared secret). These changes are logged during the following configuration load operations:

```
load merge
load replace
load override
load update
```

For more information, see the *Secure Configuration Guide for Common Criteria and Junos-FIPS*.

Related Documentation

- Examples: Loading a Configuration from a File on page 132

Examples: Loading a Configuration from a File

Figure 6: Overriding the Current Configuration

Current configuration:

```
interfaces {
 lo0 {
  unit 0 {
   family inet {
    address 127.0.0.1;
   }
  }
 }
 so-3/0/0 {
  unit 0 {
   family inet {
    address 204.69.248.181/28:
   }
  }
 }
}
```

File contents:

```
interfaces {
replace:
 so-3/0/0 {
  unit 0 {
   family inet {
    address 10.0.0.1/8;
   }
  }
 }
}
```

load override →

New contents:

```
interfaces {
 so-3/0/0 {
  unit 0 {
   family inet {
    address 10.0.0.1/8;
   }
  }
 }
}
```

1628

Copyright © 2014, Juniper Networks, Inc.

Figure 7: Using the replace Option

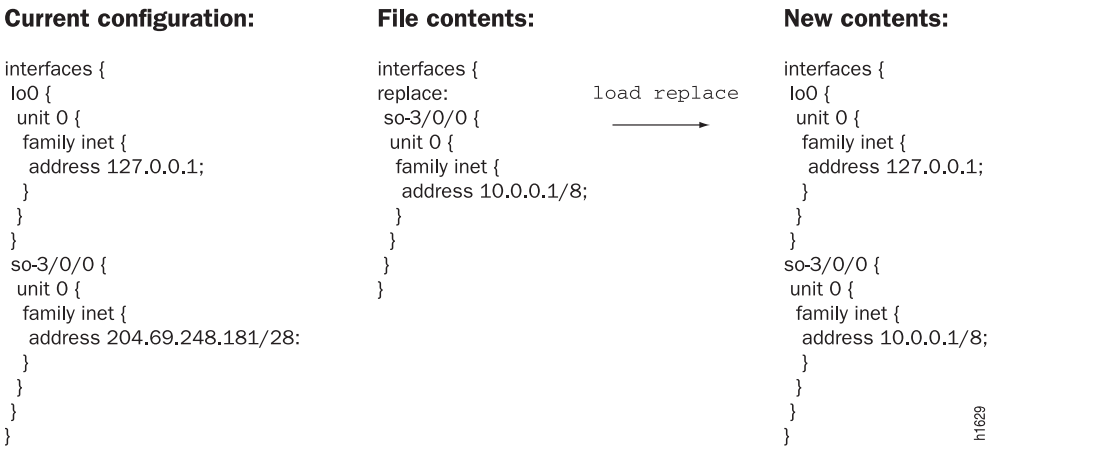
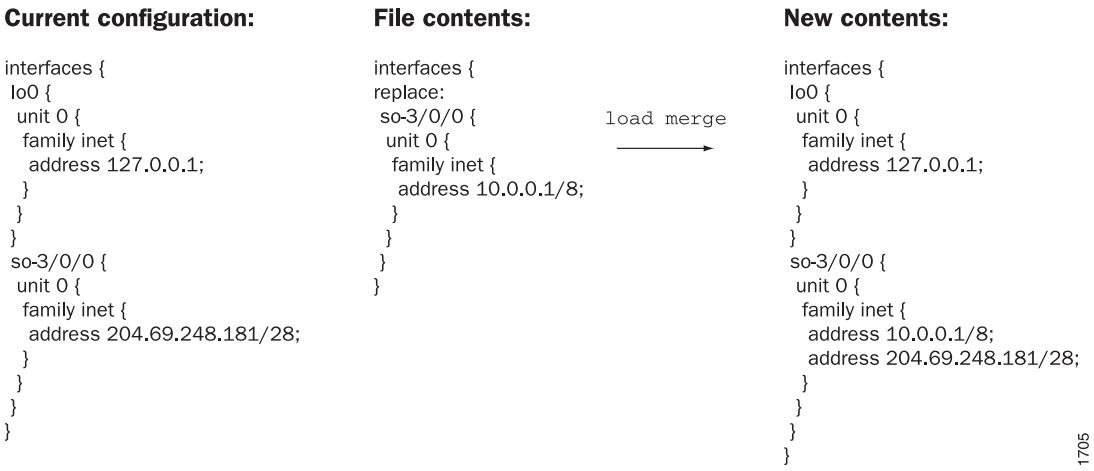


Figure 8: Using the merge Option



Copyright © 2014, Juniper Networks, Inc.

Figure 9: Using a Patch File

Current configuration:

```
interfaces {
 fxp0 {
  unit 0 {
   family inet {
    address 192.168.6.193/24;
   }
  }
 }
lo0 {
  unit 0 {
   family inet {
    address 127.0.0.1/32;
   }
  }
 }
}
```

File contents:

```
{edit interfaces}
+ so-0/0/0 {
+  unit 0 {
+   family inet {
+    address 10.0.0.1/8;
+   }
+  }
+ }
```

load patch →

New contents:

```
interfaces {
 so-0/0/0 {
  unit 0 {
   family inet {
    address 10.0.0.1/8;
   }
  }
 }
fxp0 {
  unit 0 {
   family inet {
    address 192.168.6.193/24;
   }
  }
 }
lo0 {
 unit 0 {
  family inet {
   address 127.0.0.1/32;
   }
  }
 }
}
```

h1969

Figure 10: Using the set Option

File contents:

```
edit access
set profile p1 client cl ike
edit profile p1 client cl ike
set pre-shared-key ascii-text "abcd"
set allowed-proxy-pair local 1.1.1.1 remote 2.2.2.2
exit
deactivate profile p1
top
edit system
set radius-server 1.1.1.1
```

load set ↓

New contents:

```
system {
   radius-server {
      1.1.1.1;
   }
}
access {
   inactive: profile p1 {
      client cl {
         ike {
            allowed-proxy-pair local 1.1.1.1/32 remote 2.2.2.2/32;
            pre-shared-key ascii-text "$9$Ydg4ZDjqf5FVw"; ## SECRET-DATA
         }
      }
   }
}
```

g017215

Related Documentation

- Loading a Configuration from a File on page 129

Copyright © 2014, Juniper Networks, Inc.

Creating and Returning to a Rescue Configuration

A *rescue configuration* allows you to define a known working configuration or a configuration with a known state that you can roll back to at any time. This alleviates the necessity of having to remember the rollback number with the **rollback** command. You use the rescue configuration when you need to roll back to a known configuration or as a last resort if your router or switch configuration and the backup configuration files become damaged beyond repair.

To save the most recently committed configuration as the rescue configuration so that you can return to it at any time, issue the **request system configuration rescue save** command:

```
user@host> request system configuration rescue save
```

To return to the rescue configuration, use the **rollback rescue** configuration mode command:

```
[edit]
user@host# rollback rescue
load complete
```

NOTE: If the rescue configuration does not exist, or if the rescue configuration is not a complete, viable configuration, the rollback command fails, an error message appears, and the current configuration remains active.

To activate the rescue configuration that you have loaded, use the **commit** command:

```
[edit]
user@host# rollback rescue
load complete
[edit]
user@host# commit
```

To delete an existing rescue configuration, issue the **request system configuration rescue delete** command:

```
user@host> request system configuration rescue delete
user@host>
```

For more information about the **request system configuration rescue delete** and **request system configuration rescue save** commands, see the CLI Explorer.

Related Documentation

- Comparing Configuration Changes with a Prior Version on page 118
- Saving a Configuration to a File on page 124

Compressing the Current Configuration File

By default, the current operational configuration file is compressed, and is stored in the file **juniper.conf.gz**, in the **/config** file system, along with the last three committed versions

Copyright © 2014, Juniper Networks, Inc.

of the configuration. If you have large networks, the current configuration file might exceed the available space in the **/config** file system. Compressing the current configuration file enables the file to fit in the file system, typically reducing the size of the file by 90 percent. You might want to compress your current operation configuration files when they reach 3 megabytes (MB) in size.

When you compress the current configuration file, the names of the configuration files change. To determine the size of the files in the **/config** file system, issue the **file list /config detail** command.

NOTE: **We recommend that you compress the configuration files (this is the default) to minimize the amount of disk space that they require.**

- If you want to compress the current configuration file, include the **compress-configuration-files** statement at the **[edit system]** hierarchy level:

  ```
  [edit system]
  compress-configuration-files;
  ```

  Commit the current configuration file to include the **compression-configuration-files** statement. Commit the configuration again to compress the current configuration file:

  ```
  [edit system]
  user@host# set compress-configuration-files
  user@host# commit
  commit complete
  user@host# commit
  commit complete
  ```

- If you do not want to compress the current operational configuration file, include the **no-compress-configuration-files** statement at the **[edit system]** hierarchy level:

  ```
  [edit system]
  no-compression-configuration-files;
  ```

Commit the current configuration file to include the **no-compress-configuration-files** statement. Commit the configuration again to uncompress the current configuration file:

```
[edit system]
user@host# commit
commit complete
user@host# commit
commit complete
```

Related Documentation

- Junos OS Commit Model for Router or Switch Configuration on page 97
- *compress-configuration-files*

Copyright © 2014, Juniper Networks, Inc.

Example: Protecting the Junos OS Configuration from Modification or Deletion

This example shows how to use the **protect** and **unprotect** commands in the configuration mode to protect and unprotect the CLI configuration.

- Requirements on page 137
- Overview on page 137
- Protecting a Parent-Level Hierarchy on page 138
- Protecting a Child Hierarchy on page 138
- Protecting a Configuration Statement Within a Hierarchy on page 138
- Protecting a List of Identifiers for a Configuration Statement on page 139
- Protecting an Individual Member from a Homogenous List on page 139
- Unprotecting a Configuration on page 140
- Verification on page 140

Requirements

This example uses the following hardware and software components:

- A J Series, M Series, MX Series, or T Series device
- Junos OS 11.2 or later running on all devices

Overview

The Junos OS enables you to protect the device configuration from being modified or deleted by other users. This can be accomplished by using the **protect** command in the configuration mode of the CLI. Likewise, you can also unprotect a protected configuration by using the **unprotect** command.

These commands can be used at any level of the configuration hierarchy—a top-level parent hierarchy or a configuration statement or an identifier within the lowest level of the hierarchy.

If a configuration hierarchy is protected, users cannot perform the following activities:

- Deleting or modifying a hierarchy or a statement or identifier within the protected hierarchy
- Inserting a new configuration statement or an identifier within the protected hierarchy
- Renaming a statement or identifier within the protected hierarchy
- Copying a configuration into a protected hierarchy
- Activating or deactivating statements within a protected hierarchy
- Annotating a protected hierarchy

Copyright © 2014, Juniper Networks, Inc.

Protecting a Parent-Level Hierarchy

Step-by-Step Procedure

To protect a configuration at the top level of the hierarchy:

- Identify the hierarchy that you want to protect and issue the **protect** command for the hierarchy at the **[edit]** hierarchy level.

  For example, if you want to protect the entire **[edit access]** hierarchy level, issue the following command:

  ```
  [edit]
  user@host# protect access
  ```

Results Protects all elements under the parent hierarchy.

NOTE:

- If you issue the protect command for a hierarchy that is not used in the configuration, the Junos OS CLI displays the following error message:

  ```
  [edit]
  user@host# protect access
  warning: statement not found
  ```

Protecting a Child Hierarchy

Step-by-Step Procedure

To protect a child hierarchy contained within a parent hierarchy:

- Navigate to the parent container hierarchy. Use the **protect** command for the hierarchy at the parent level.

  For example, if you want to protect the **[edit system syslog console]** hierarchy level, use the following command at the **[edit system syslog]** hierarchy level.

  ```
  [edit system syslog]
  user@host# protect console
  ```

Results Protects all elements under the child hierarchy.

Protecting a Configuration Statement Within a Hierarchy

Step-by-Step Procedure

To protect a configuration statement within a hierarchy level:

- Navigate to the hierarchy level containing the statement that you want to protect and issue the **protect** command for the hierarchy.

  For example, if you want to protect the **host-name** statement under the **[edit system]** hierarchy level, issue the following command:

  ```
  [edit system]
  user@host# protect host-name
  ```


Copyright © 2014, Juniper Networks, Inc.

Protecting a List of Identifiers for a Configuration Statement

Step-by-Step Procedure

Some configuration statements can take multiple values. For example, the **address** statement at the **[edit system login deny-sources]** hierarchy level can take a list of hostnames, IPv4 addresses, or IPv6 addresses. Suppose you have the following configuration:

```
[edit system login]
deny-sources {
  address [ 172.17.28.19 172.17.28.20 172.17.28.21 172.17.28.22];
}
```

- To protect all the addresses for the **address** statement, issue the following command at the **[edit]** level:

```
[edit]
user@host# protect system login deny-sources address
```

Results

All the addresses ([172.17.28.19 172.17.28.20 172.17.28.21 172.17.28.22]) for the **address** statement are protected.

Protecting an Individual Member from a Homogenous List

Step-by-Step Procedure

Suppose you have the following configuration:

```
[edit groups ]
test1 {
  system {
    name-server {
      10.1.2.1;
      10.1.2.2;
      10.1.2.3;
      10.1.2.4;
    }
  }
}
```

- To protect one or more individual addresses for the **name-server** statement, issue the following command at the **[edit]** level:

```
[edit]
user@host# protect groups test1 system name-server 10.1.2.1
user@host# protect groups test1 system name-server 10.1.2.4
```

Results

Addresses 10.1.2.1 and 10.1.2.4 are protected.

Copyright © 2014, Juniper Networks, Inc.

Unprotecting a Configuration

Step-by-Step Procedure

Suppose you have the following configuration at the **[edit system]** hierarchy level:

```
protect: system {
    host-name bigping;
    domain-search 10.1.2.1;
    login {
        deny-sources {
            protect: address [ 172.17.28.19 172.17.28.173 172.17.28.0 174.0.0.0 ];
        }
    }
}
```

- To unprotect the entire **[edit system]** hierarchy level, issue the following command at the **[edit]** level:

```
[edit]
user@host# unprotect system
```

Results

The entire **system** hierarchy level is unprotected.

Verification

Verify That a Hierarchy Is Protected Using the show Command

Purpose

To check that a configuration hierarchy is protected.

Action

In the configuration mode, issue the **show** command at the **[edit]** hierarchy level to see all the configuration hierarchies and configuration statements that are protected.

NOTE: All protected hierarchies or statements are prefixed with a **protect:** string.

```
...
protect: system {
    host-name bigping;
    domain-search 10.1.2.1;
    login {
        deny-sources {
            protect: address [ 172.17.28.19 172.17.28.173 172.17.28.0 174.0.0.0 ];
        }
    }
}
...
```

Verify That a Hierarchy Is Protected by Attempting to Modify a Configuration

Purpose

To verify that a configuration is protected by trying to modify the configuration using the **activate**, **copy**, **insert**, **rename**, and **delete** commands.

Copyright © 2014, Juniper Networks, Inc.

Action To verify that a configuration is protected:

1. Try using the **activate**, **copy**, **insert**, **rename**, and **delete** commands for a top-level hierarchy or a child-level hierarchy or a statement within the hierarchy.

   For a protected hierarchy or statement, the Junos OS displays an appropriate warning that the command has not executed. For example:

   ```
   protect: system {
     host-name a;
     inactive: domain-search [ a b ];
   }
   ```

2. To verify that the hierarchy is protected, try issuing the **activate** command for the **domain-search** statement:

   [edit system]

   ```
   user@host# activate system domain-search
   ```

   The Junos OS CLI displays an appropriate message:

   ```
   warning: [system] is protected, 'system domain-search' cannot be activated
   ```

Verify Usage of the protect Command

Purpose To view the **protect** commands used for protecting a configuration.

Action

1. Navigate to the required hierarchy.
2. Issue the **show | display set relative** command.

```
user@host> show | display set relative
set system host-name bigping
set system domain-search 10.1.2.1
set system login deny-sources address 172.17.28.19
set system login deny-sources address 172.17.28.173
set system login deny-sources address 172.17.28.0
set system login deny-sources address 174.0.0.0
protect system login deny-sources address
protect system
```

View the Configuration in XML

Purpose To check if the protected hierarchies or statements are also displayed in the XML. Protected hierarchies, statements, or identifiers are displayed with the **| display xml** attribute in the XML.

Copyright © 2014, Juniper Networks, Inc.

Action To view the configuration in XML:

Copyright © 2014, Juniper Networks, Inc.

1. Navigate to the hierarchy you want to view and issue the **show** command with the pipe symbol and option **| display xml**:

   [edit system]

```
user@host# show | display xml
[edit]
user@host# show system | display xml
<rpc-reply xmlns:junos="http://xml.juniper.net/junos/11.2I0/junos">
    <configuration junos:changed-seconds="1291279234"
junos:changed-localtime="2010-12-02 00:40:34 PST">
            <system protect="protect">
                <host-name>bigping</host-name>
                <domain-search>10.1.2.1</domain-search>
                <login>
                    <message>


        \jnpr

        \tUNAUTHORIZED USE OF THIS ROUTER
        \tIS STRICTLY PROHIBITED!


    </message>
                <class>
                    <name>a</name>
                    <allow-commands>commit-synchronize</allow-commands>
                    <deny-commands>commit</deny-commands>
                </class>
                <deny-sources>
                    <address protect="protect">172.17.28.19</address>
                    <address protect="protect">172.17.28.173</address>
                    <address protect="protect">172.17.28.0</address>
                    <address protect="protect">174.0.0.0</address>
                </deny-sources>
            </login>
            <syslog>
                <archive>
                </archive>
            </syslog>
        </system>
</configuration>
<cli>
    <banner>[edit]</banner>
</cli>
    </rpc-reply>
```

NOTE: Loading an XML configuration with the unprotect="unprotect" tag unprotects an already protected hierarchy. For example, suppose you load the following XML hierarchy:

```
<protocols unprotect="unprotect">
 <ospf>
     <area>
  <name>0.0.0.0</name>
  <interface>
      <name>all</name>
  </interface>
```

Copyright © 2014, Juniper Networks, Inc.

```
    </area>
  </ospf>
    </protocols>
```

The **[edit protocols]** hierarchy becomes unprotected if it is already protected.

Synchronizing Routing Engines

If your router has two Routing Engines, you can manually direct one Routing Engine to synchronize its configuration with the other by issuing the **commit synchronize** command. The Routing Engine on which you execute this command (requesting Routing Engine) copies and loads its candidate configuration to the other (responding Routing Engine). Both Routing Engines then perform a syntax check on the candidate configuration file being committed. If no errors are found, the configuration is activated and becomes the current operational configuration on both Routing Engines.

The **commit synchronize** command does not work if the responding Routing Engine has uncommitted configuration changes. However, you can enforce commit synchronization on the Routing Engines by using the **force** option. When you issue the **commit synchronize** command with the **force** option from one Routing Engine, the configuration sessions on the other Routing Engine will be terminated and its configuration synchronized with that on the Routing Engine from which you issued the command.

NOTE: We recommend that you use the force option only if you are unable to resolve the issues that caused the commit synchronize command to fail.

For example, if you are logged in to **re1 (**requesting Routing Engine**)** and you want **re0** (responding Routing Engine) to have the same configuration as **re1**, issue the **commit synchronize** command on **re1**. **re1** copies and loads its candidate configuration to **re0**. Both Routing Engines then perform a syntax check on the candidate configuration file being committed. If no errors are found, **re1**'s candidate configuration is activated and becomes the current operational configuration on both Routing Engines.

NOTE: When you issue the commit synchronize command, you must use the groups re0 and re1. For information about how to use the apply-groups statement, see "Applying a Junos OS Configuration Group" on page 201.

The responding Routing Engine must be running Junos OS Release 5.0 or later.

For information about issuing the commit synchronize command on a routing matrix, see the *Junos OS Administration Library for Routing Devices*.

Copyright © 2014, Juniper Networks, Inc.

To synchronize a Routing Engine's current operational configuration file with the other, log in to the Routing Engine from which you want to synchronize and issue the **commit synchronize** command:

```
[edit]
user@host# commit synchronize
re0:
configuration check succeeds
re1:
commit complete
re0:
commit complete
```

NOTE: You can also add the **commit synchronize** statement at the **[edit system]** hierarchy level so that a **commit** command automatically invokes a **commit synchronize** command by default. For more information, see the *Junos OS Administration Library for Routing Devices*.

To enforce a commit synchronize on the Routing Engines, log in to the Routing Engine from which you want to synchronize and issue the **commit synchronize** command with the **force** option:

```
[edit]
user@host# commit synchronize force
re0:
re1:
commit complete
re0:
commit complete
[edit]
user@host#
```

NOTE:

- If you have nonstop routing enabled on your router, you must enter the **commit synchronize** command from the master Routing Engine after you make any changes to the configuration. If you enter this command on the backup Routing Engine, the Junos OS displays a warning and commits the configuration.
- Starting with Junos OS Release 9.3, accounting of backup Routing Engine events or operations is not supported on accounting servers such as TACACS+ or RADIUS. Accounting is only supported for events or operations on a master Routing Engine.

For the commit synchronization process, the master Routing Engine commits the configuration and sends a copy of the configuration to the backup Routing Engine. Then the backup Routing Engine loads and commits the configuration. So, the commit synchronization between the master and backup Routing Engines takes place one Routing

Copyright © 2014, Juniper Networks, Inc.

Engine at a time. If the configuration has a large text size or many apply-groups, commit times can be longer than desired.

You can use the **commit fast-synchronize** statement to have the synchronization between the master and backup Routing Engines occur simultaneously instead of sequentially. This can reduce the time needed for synchronization because the commits on the master and backup Routing Engines occur in parallel.

Include the **fast-synchronize** statement at the **[edit system]** hierarchy level to have the synchronization occur simultaneously between the master and the backup Routing Engines:

```
[edit system]
commit fast-synchronize;
```

NOTE:

- When the **fast-synchronize** statement is configured, the commits on the master Routing Engine and the backup Routing Engine run in parallel. In this process, the configuration is validated only on the Routing Engine where you execute the **commit** command. Therefore, it is recommended not to include too many configuration details in groups like re0 and re1, because the configuration specified in group re0 is applied only if the current Routing Engine is in slot 0. Likewise, the configuration specified in group re1 is applied only if the current Routing Engine is in slot 1.
- Ensure that the Junos OS software version running on both the Routing Engines is same.

You can use the **commit synchronize scripts** command to synchronize a Routing Engine's configuration and all commit, event, lib and op scripts with the other Routing Engine. If the **load-scripts-from-flash** statement is configured for the requesting Routing Engine, the device synchronizes the scripts from flash memory on the requesting Routing Engine to flash memory on the responding Routing Engine. Otherwise, the device synchronizes the scripts from the hard disk on the requesting Routing Engine to the hard disk on the responding Routing Engine. The device synchronizes all scripts regardless of whether they are enabled in the configuration or have been updated since the last synchronization.

To synchronize a Routing Engine's configuration file and all commit, event, lib, and op scripts with the other Routing Engine, log in to the Routing Engine from which you want to synchronize, and issue the **commit synchronize scripts** command:

```
[edit]
user@host# commit synchronize scripts
re0:
configuration check succeeds
re1:
commit complete
re0:
commit complete
```


Copyright © 2014, Juniper Networks, Inc.

If the commit check operation fails for the requesting Routing Engine, the process stops, and the scripts are not copied to the responding Routing Engine. If the commit check or commit operation fails for the responding Routing Engine, the scripts are still synchronized, since the synchronization occurs prior to the commit check operation on the responding Routing Engine.

Include the **synchronize** statement at the **[edit system scripts]** hierarchy level to synchronize scripts every time you issue a **commit synchronize** command.

```
[edit system scripts]
synchronize;
```

NOTE:

- If commit fails on either Routing Engine, the commit process is rolled back on the other Routing Engine as well. This ensures that both Routing Engines have the same configuration.
- When the **fast-synchronize** statement is configured, the commits on the master Routing Engine and the backup Routing Engine run in parallel. In this process, the configuration is validated only on the Routing Engine where you execute the **commit** command. Therefore, it is recommended not to include too many configuration details in groups like re0 and re1, because the configuration specified in group re0 is applied only if the current Routing Engine is in slot 0. Likewise, the configuration specified in group re1 is applied only if the current Routing Engine is in slot 1.
- Ensure that the Junos OS software version running on both the Routing Engines is same.

Related Documentation

- *Configuring the Junos OS to Support Redundancy on Routers Having Multiple Routing Engines or Switching Boards*
- *Junos OS Routing Engine Components and Processes*
- *Configuring Junos OS for the First Time on a Device with Dual Routing Engines*

Configuring Multiple Routing Engines to Synchronize Committed Configurations Automatically

If your router or switch has multiple Routing Engines, you can manually direct one Routing Engine to synchronize its configuration with the others by issuing the **commit synchronize** command.

To make the Routing Engines synchronize automatically whenever a configuration is committed, include the **commit synchronize** statement at the **[edit system]** hierarchy level:

```
[edit system]
commit synchronize;
```

Copyright © 2014, Juniper Networks, Inc.

The Routing Engine on which you execute the **commit** command (requesting Routing Engine) copies and loads its candidate configuration to the other (responding) Routing Engines. All Routing Engines then perform a syntax check on the candidate configuration file being committed. If no errors are found, the configuration is activated and becomes the current operational configuration on all Routing Engines.

For the commit synchronization process, the master Routing Engine commits the configuration and sends a copy of the configuration to the backup Routing Engine. Then the backup Routing Engine loads and commits the configuration. So, the commit synchronization between the master and backup Routing Engines takes place one Routing Engine at a time. If the configuration has a large text size or many apply-groups, commit times can be longer than desired.

You can use the **commit fast-synchronize** statement to have the synchronization between the master and backup Routing Engines occur simultaneously instead of sequentially. This can reduce the time needed for synchronization because the commits on the master and backup Routing Engines occur in parallel.

Include the **fast-synchronize** statement at the **[edit system]** hierarchy level to have synchronize occur simultaneously between the master and the backup Routing Engines:

```
[edit system]
commit fast-synchronize
```

NOTE:

- If commit fails on either Routing Engine, the commit process is rolled back on the other Routing Engine as well. This ensures that both Routing Engines have the same configuration.
- When the **fast-synchronize** statement is configured, the commits on the master Routing Engine and the backup Routing Engine run in parallel. In this process, the configuration is validated only on the Routing Engine where you execute the **commit** command. Therefore, it is recommended not to include too many configuration details in groups like re0 and re1, because the configuration specified in group re0 is applied only if the current Routing Engine is in slot 0. Likewise, the configuration specified in group re1 is applied only if the current Routing Engine is in slot 1.
- Ensure that the Junos OS software version running on both the Routing Engines is same.

Related Documentation

- Junos OS Commit Model for Router or Switch Configuration on page 97

Copyright © 2014, Juniper Networks, Inc.

CHAPTER 7

Using Operational Commands to Monitor a Device

- Overview of Junos OS CLI Operational Mode Commands on page 149
- Junos OS Operational Mode Commands That Combine Other Commands on page 152
- Understanding the Brief, Detail, Extensive, and Terse Options of Junos OS Operational Commands on page 153
- Controlling the Scope of an Operational Mode Command on page 154
- Monitoring Who Uses the Junos OS CLI on page 157
- Interface Naming Conventions Used in the Junos OS Operational Commands on page 158
- Viewing Files and Directories on a Device Running Junos OS on page 159
- Displaying Junos OS Information on page 163
- Managing Programs and Processes Using Junos OS Operational Mode Commands on page 165
- Using the Junos OS CLI Comment Character # for Operational Mode Commands on page 170
- Example: Using Comments in Junos OS Operational Mode Commands on page 170

Overview of Junos OS CLI Operational Mode Commands

This topic provides an overview of Junos OS CLI operational mode commands and contains the following sections:

- CLI Command Categories on page 149
- Commonly Used Operational Mode Commands on page 151

CLI Command Categories

When you log in to a device running Junos OS and the CLI starts, there are several broad groups of CLI commands:

Copyright © 2014, Juniper Networks, Inc.

- Commands for controlling the CLI environment—Some set commands in the **set** hierarchy configure the CLI display screen. For information about these commands, see "Understanding the Junos OS CLI Modes, Commands, and Statement Hierarchies" on page 25.
- Commands for monitoring and troubleshooting—The following commands display information and statistics about the software and test network connectivity. Detailed command descriptions are provided in the *Junos OS Interfaces Command Reference*.
  - **clear**—Clear statistics and protocol database information.
  - **mtrace**—Trace mtrace packets from source to receiver.
  - **monitor**—Perform real-time debugging of various software components, including the routing protocols and interfaces.
  - **ping**—Determine the reachability of a remote network host.
  - **show**—Display the current configuration and information about interfaces, routing protocols, routing tables, routing policy filters, system alarms, and the chassis.
  - **test**—Test the configuration and application of policy filters and autonomous system (AS) path regular expressions.
  - **traceroute**—Trace the route to a remote network host.
- Commands for connecting to other network systems—The **ssh** command opens Secure Shell connections, and the **telnet** command opens telnet sessions to other hosts on the network. For information about these commands, see the CLI Explorer.
- Commands for copying files—The **copy** command copies files from one location on the router or switch to another, from the router or switch to a remote system, or from a remote system to the router or switch. For information about these commands, see the CLI Explorer.
- Commands for restarting software processes—The commands in the **restart** hierarchy restart the various Junos OS processes, including the routing protocol, interface, and SNMP. For information about these commands, see the CLI Explorer.
- A command—**request**—for performing system-level operations, including stopping and rebooting the router or switch and loading Junos OS images. For information about this command, see the CLI Explorer.
- A command—**start**—to exit the CLI and start a UNIX shell. For information about this command, see the CLI Explorer.
- A command—**configure**—for entering configuration mode, which provides a series of commands that configure Junos OS, including the routing protocols, interfaces, network management, and user access. For information about the CLI configuration commands, see "Understanding Junos OS CLI Configuration Mode" on page 58.
- A command—**quit**—to exit the CLI. For information about this command, see the CLI Explorer.
- For more information about the CLI operational mode commands, see the CLI Explorer.
  .

 Copyright © 2014, Juniper Networks, Inc.

Commonly Used Operational Mode Commands

Table 8 on page 151 lists some operational commands you may find useful for monitoring router or switch operation. For a complete description of operational commands, see the Junos OS command references.

NOTE: The QFX3500 switch does not support the IS-IS, OSPF, BGP, MPLS, and RSVP protocols.

Table 8: Commonly Used Operational Mode Commands

Items to Check	Description	Command
Software version	Versions of software running on the router or switch	**show version**
Log files	Contents of the log files	**monitor**
	Log files and their contents and recent user logins	**show log**
Remote systems	Host reachability and network connectivity	**ping**
	Route to a network system	**traceroute**
Configuration	Current system configuration	**show configuration**
Manipulate files	List of files and directories on the router or switch	**file list**
	Contents of a file	**file show**
Interface information	Detailed information about interfaces	**show interfaces**
Chassis	Chassis alarm status	**show chassis alarms**
	Information currently on craft display	**show chassis craft-interface**
	Router or switch environment information	**show chassis environment**
	Hardware inventory	**show chassis hardware**
Routing table information	Information about entries in the routing tables	**show route**
Forwarding table information	Information about data in the kernel's forwarding table	**show route forwarding-table**
IS-IS	Adjacent routers or switches	**show isis adjacency**
OSPF	Display standard information about OSPF neighbors	**show ospf neighbor**
BGP	Display information about BGP neighbors	**show bgp neighbor**

Copyright © 2014, Juniper Networks, Inc.

Table 8: Commonly Used Operational Mode Commands *(continued)*

Items to Check	Description	Command
MPLS	Status of interfaces on which MPLS is running	**show mpls interface**
	Configured LSPs on the router or switch, as well as all ingress, transit, and egress LSPs	**show mpls lsp**
	Routes that form a label-switched path	**show route label-switched-path**
RSVP	Status of interfaces on which RSVP is running	**show rsvp interface**
	Currently active RSVP sessions	**show rsvp session**
	RSVP packet and error counters	**show rsvp statistics**

Related Documentation

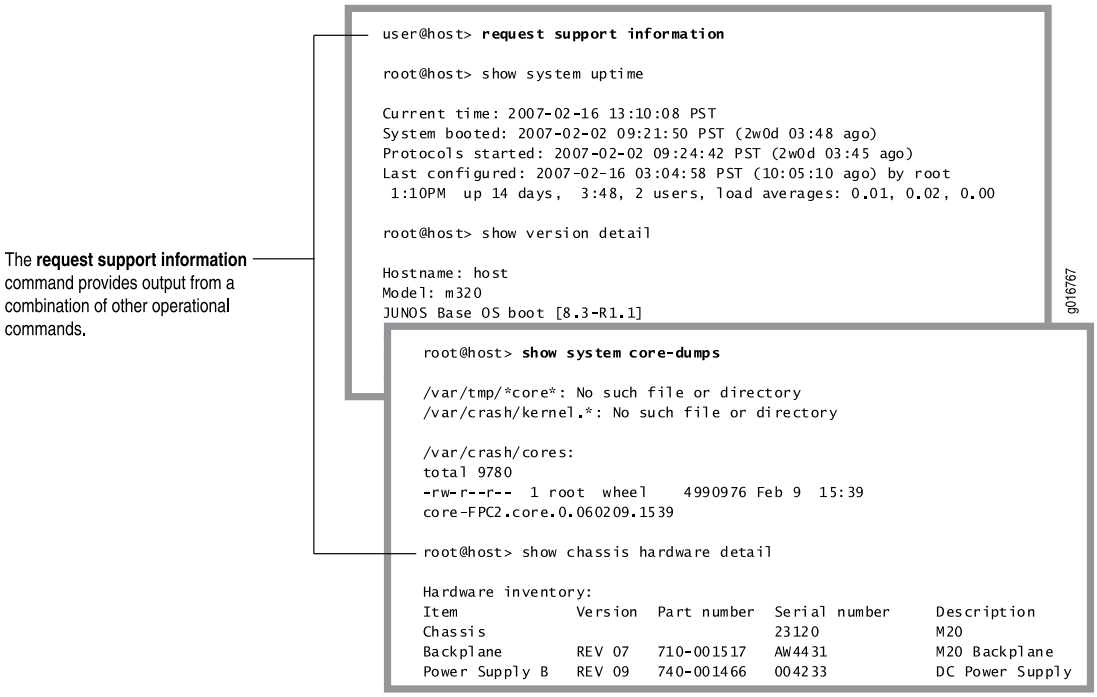
- Junos OS Operational Mode Commands That Combine Other Commands on page 152
- Understanding the Brief, Detail, Extensive, and Terse Options of Junos OS Operational Commands on page 153

Junos OS Operational Mode Commands That Combine Other Commands

In some cases, some Junos OS operational commands are created from a combination of other operational commands. These commands can be useful shortcuts for collecting information about the device, as shown in Figure 11 on page 153.

Copyright © 2014, Juniper Networks, Inc.

Figure 11: Commands That Combine Other Commands


The request support information command provides output from a combination of other operational commands.
user@host> request support information
root@host> show system uptime
Current time: 2007-02-16 13:10:08 PST
System booted: 2007-02-02 09:21:50 PST (2w0d 03:48 ago)
Protocols started: 2007-02-02 09:24:42 PST (2w0d 03:45 ago)
Last configured: 2007-02-16 03:04:58 PST (10:05:10 ago) by root
1:10PM up 14 days, 3:48, 2 users, load averages: 0.01, 0.02, 0.00
root@host> show version detail
Hostname: host
Model: m320
JUNOS Base OS boot [8.3-R1.1]
root@host> show system core-dumps
/var/tmp/*core*: No such file or directory
/var/crash/kernel.*: No such file or directory
/var/crash/cores:
total 9780
-rw-r--r-- 1 root wheel 4990976 Feb 9 15:39
core-FPC2.core.0.060209.1539
root@host> show chassis hardware detail
Hardware inventory:
Item Version Part number Serial number Description
Chassis 23120 M20
Backplane REV 07 710-001517 AW4431 M20 Backplane
Power Supply B REV 09 740-001466 004233 DC Power Supply
g016767

Related Documentation

- Overview of Junos OS CLI Operational Mode Commands on page 149
- Understanding the Brief, Detail, Extensive, and Terse Options of Junos OS Operational Commands on page 153

Understanding the Brief, Detail, Extensive, and Terse Options of Junos OS Operational Commands

The Junos OS operational mode commands can include **brief**, **detail**, **extensive**, or **terse** options. You can use these options to control the amount of information you want to view.

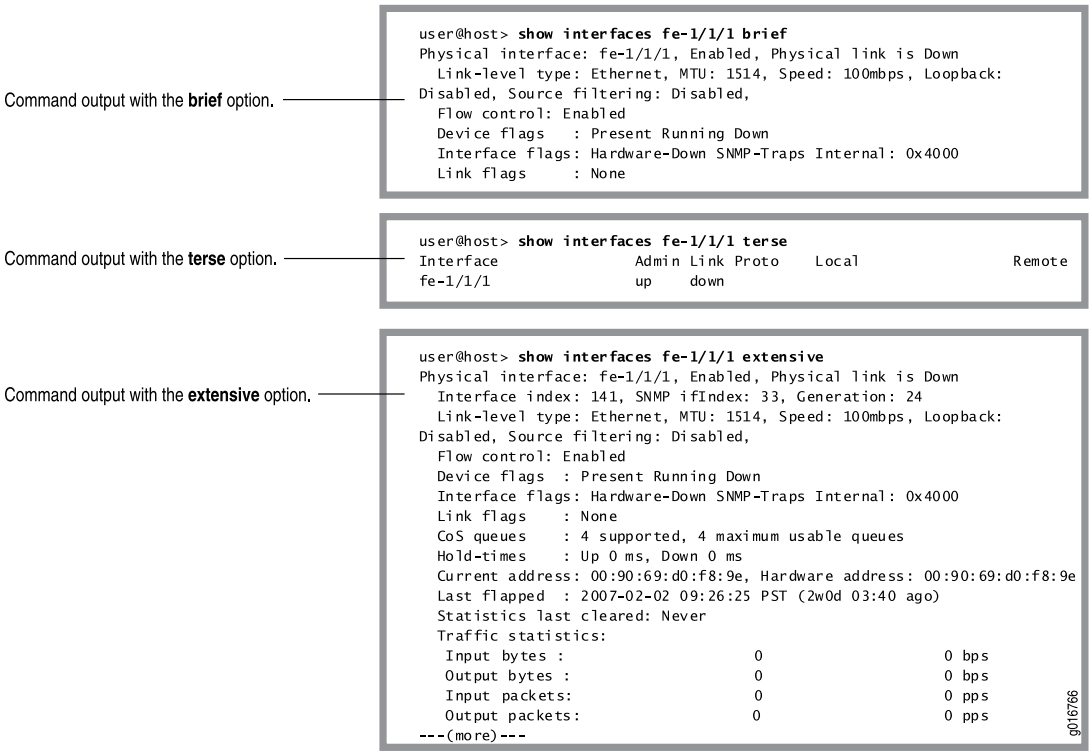
1. Use the ? prompt to list options available for the command. For example:

```
user@host> show interfaces fe-1/1/1 ?
Possible completions:
<[Enter]>            Execute this command
  brief              Display brief output
  descriptions       Display interface description strings
  detail             Display detailed output
  extensive          Display extensive output
  media              Display media information
  snmp-index         SNMP index of interface
  statistics         Display statistics and detailed output
  terse              Display terse output
  |                  Pipe through a command
```

2. Choose the option you wish to use with the command. (See Figure 12 on page 154.)

Copyright © 2014, Juniper Networks, Inc.

Figure 12: Command Output Options


Command output with the brief option.
user@host> show interfaces fe-1/1/1 brief
Physical interface: fe-1/1/1, Enabled, Physical link is Down
Link-level type: Ethernet, MTU: 1514, Speed: 100mbps, Loopback:
Disabled, Source filtering: Disabled,
Flow control: Enabled
Device flags : Present Running Down
Interface flags: Hardware-Down SNMP-Traps Internal: 0x4000
Link flags : None
Command output with the terse option.
user@host> show interfaces fe-1/1/1 terse
Interface Admin Link Proto Local Remote
fe-1/1/1 up down
Command output with the extensive option.
user@host> show interfaces fe-1/1/1 extensive
Physical interface: fe-1/1/1, Enabled, Physical link is Down
Interface index: 141, SNMP ifIndex: 33, Generation: 24
Link-level type: Ethernet, MTU: 1514, Speed: 100mbps, Loopback:
Disabled, Source filtering: Disabled,
Flow control: Enabled
Device flags : Present Running Down
Interface flags: Hardware-Down SNMP-Traps Internal: 0x4000
Link flags : None
CoS queues : 4 supported, 4 maximum usable queues
Hold-times : Up 0 ms, Down 0 ms
Current address: 00:90:69:d0:f8:9e, Hardware address: 00:90:69:d0:f8:9e
Last flapped : 2007-02-02 09:26:25 PST (2w0d 03:40 ago)
Statistics last cleared: Never
Traffic statistics:
Input bytes : 0 0 bps
Output bytes : 0 0 bps
Input packets: 0 0 pps
Output packets: 0 0 pps
---(more)---
g016766

Related Documentation

- Overview of Junos OS CLI Operational Mode Commands on page 149
- Controlling the Scope of an Operational Mode Command on page 154

Controlling the Scope of an Operational Mode Command

The Junos OS CLI operational commands include options that you can use to identify specific components on a device running Junos OS. For example:

1. Type the **show interfaces** command to display information about all interfaces on the router.

```
user@host> show interfaces
Physical interface: so-0/0/0, Enabled, Physical link is Up
  Interface index: 128, SNMP ifIndex: 23
  Link-level type: PPP, MTU: 4474, Clocking: Internal, SONET mode, Speed: OC3,

  Loopback: None, FCS: 16, Payload scrambler: Enabled
  Device flags   : Present Running
  Interface flags: Point-To-Point SNMP-Traps Internal: 0x4000
  Link flags     : Keepalives
  Keepalive settings: Interval 10 seconds, Up-count 1, Down-count 3
  Keepalive: Input: 13861 (00:00:05 ago), Output: 13891 (00:00:01 ago)
  LCP state: Opened
  NCP state: inet: Opened, inet6: Not-configured, iso: Opened, mpls:
Not-configured
  CHAP state: Closed
  PAP state: Closed
```


Copyright © 2014, Juniper Networks, Inc.

```
  CoS queues     : 4 supported, 4 maximum usable queues
  Last flapped   : 2008-06-02 17:16:14 PDT (1d 14:21 ago)
  Input rate     : 40 bps (0 pps)
  Output rate    : 48 bps (0 pps)

---(more)---
```

2. To display information about a specific interface, type that interface as a command option:

```
user@host> show interfaces fe-0/1/3
Physical interface: fe-0/1/3, Enabled, Physical link is Up
  Interface index: 135, SNMP ifIndex: 30
  Link-level type: Ethernet, MTU: 1514, Speed: 100mbps, MAC-REWRITE Error:
None,
  Loopback: Disabled, Source filtering: Disabled, Flow control: Enabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 4 supported, 4 maximum usable queues
  Current address: 00:05:85:8f:c8:22, Hardware address: 00:05:85:8f:c8:22
  Last flapped   : 2008-06-02 17:16:15 PDT (1d 14:28 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Active alarms  : None
  Active defects : None

user@host>
```

Operational Mode Commands on a TX Matrix Router or TX Matrix Plus Router

When you issue operational mode commands on the TX Matrix router, CLI command options allow you to restrict the command output to show only a component of the routing matrix rather than the routing matrix as a whole.

These are the options shown in the CLI:

- **scc**—The TX Matrix router (or switch-card chassis)
- **sfc**—The TX Matrix Plus router (also referred to as or switch-fabric chassis)
- **lcc *number***—A specific router in a routing matrix based on a TX Matrix router or a TX Matrix Plus router.
- **all-lcc**—All T640 routers (in a routing matrix based on a TX Matrix router) or all T1600 routers or T4000 routers (in a routing matrix based on a TX Matrix Plus router).

If you specify none of these options, then the command applies by default to the whole routing matrix.

Examples of Routing Matrix Command Options

The following output samples, using the **show version** command, demonstrate some different options for viewing information about the routing matrix.

```
user@host> show version ?
Possible completions:
  <[Enter]>            Execute this command
  all-lcc              Show software version on all LCC chassis
```

Copyright © 2014, Juniper Networks, Inc.

```
  brief               Display brief output
  detail              Display detailed output
  lcc                 Show software version on specific LCC (0..3)
  scc                 Show software version on the SCC
  |                   Pipe through a command
```

Sample Output: No Routing Matrix Options Specified

```
user@host> show version
scc-re0:
--------------------------------------------------------------------------
Hostname: scc
Model: TX Matrix
JUNOS Base OS boot [7.0-20040630.0]
JUNOS Base OS Software Suite [7.0-20040629.0]
JUNOS Kernel Software Suite [7.0-20040630.0]
JUNOS Packet Forwarding Engine Support (T-Series) [7.0-20040630.0]
JUNOS Routing Software Suite [7.0-20040630.0]
JUNOS Online Documentation [7.0-20040630.0]
JUNOS Crypto Software Suite [7.0-20040630.0]
lcc0-re0:
--------------------------------------------------------------------------
Hostname: lcc0
Model: t640
JUNOS Base OS boot [7.0-20040630.0]
JUNOS Base OS Software Suite [7.0-20040629.0]
JUNOS Kernel Software Suite [7.0-20040630.0]
JUNOS Packet Forwarding Engine Support (T-Series) [7.0-20040630.0]
JUNOS Routing Software Suite [7.0-20040630.0]
JUNOS Online Documentation [7.0-20040630.0]
JUNOS Crypto Software Suite [7.0-20040630.0]
JUNOS Support Tools Package [7.0-20040630.0]
lcc1-re0:
--------------------------------------------------------------------------
Hostname: lcc1
Model: t640
JUNOS Base OS boot [7.0-20040630.0]
JUNOS Base OS Software Suite [7.0-20040629.0]
JUNOS Kernel Software Suite [7.0-20040630.0]
JUNOS Packet Forwarding Engine Support (T-Series) [7.0-20040630.0]
JUNOS Routing Software Suite [7.0-20040630.0]
JUNOS Online Documentation [7.0-20040630.0]
JUNOS Crypto Software Suite [7.0-20040630.0]
JUNOS Support Tools Package [7.0-20040630.0]
```

Sample Output: TX Matrix Router Only (scc Option)

```
user@host> show version scc
Hostname: scc
Model: TX Matrix
JUNOS Base OS boot [7.0-20040630.0]
JUNOS Base OS Software Suite [7.0-20040629.0]
JUNOS Kernel Software Suite [7.0-20040630.0]
JUNOS Packet Forwarding Engine Support (T-Series) [7.0-20040630.0]
JUNOS Routing Software Suite [7.0-20040630.0]
JUNOS Online Documentation [7.0-20040630.0]
JUNOS Crypto Software Suite [7.0-20040630.0]
```

Sample Output: Specific T640 Router (lcc number Option)

```
user@host> show version lcc 0
```


Copyright © 2014, Juniper Networks, Inc.

```
lcc0-re0:
--------------------------------------------------------------------------
Hostname: lcc0
Model: t640
JUNOS Base OS boot [7.0-20040630.0]
JUNOS Base OS Software Suite [7.0-20040629.0]
JUNOS Kernel Software Suite [7.0-20040630.0]
JUNOS Packet Forwarding Engine Support (T-Series) [7.0-20040630.0]
JUNOS Routing Software Suite [7.0-20040630.0]
JUNOS Online Documentation [7.0-20040630.0]
JUNOS Crypto Software Suite [7.0-20040630.0]
JUNOS Support Tools Package [7.0-20040630.0]
```

Sample Output: All T640 Routers (all-lcc Option)

```
user@host> show version all-lcc
lcc0-re0:
--------------------------------------------------------------------------
Hostname: lcc0
Model: t640
JUNOS Base OS boot [7.0-20040630.0]
JUNOS Base OS Software Suite [7.0-20040629.0]
JUNOS Kernel Software Suite [7.0-20040630.0]
JUNOS Packet Forwarding Engine Support (T-Series) [7.0-20040630.0]
JUNOS Routing Software Suite [7.0-20040630.0]
JUNOS Online Documentation [7.0-20040630.0]
JUNOS Crypto Software Suite [7.0-20040630.0]
JUNOS Support Tools Package [7.0-20040630.0]
lcc1-re0:
--------------------------------------------------------------------------
Hostname: lcc1
Model: t640
JUNOS Base OS boot [7.0-20040630.0]
JUNOS Base OS Software Suite [7.0-20040629.0]
JUNOS Kernel Software Suite [7.0-20040630.0]
JUNOS Packet Forwarding Engine Support (T-Series) [7.0-20040630.0]
JUNOS Routing Software Suite [7.0-20040630.0]
JUNOS Online Documentation [7.0-20040630.0]
JUNOS Crypto Software Suite [7.0-20040630.0]
JUNOS Support Tools Package [7.0-20040630.0]
```

Related Documentation

- Interface Naming Conventions Used in the Junos OS Operational Commands on page 158
- Using the Junos OS CLI Comment Character # for Operational Mode Commands on page 170

Monitoring Who Uses the Junos OS CLI

Depending upon how you configure Junos OS, multiple users can log in to the router, use the CLI, and configure or modify the software configuration.

If, when you enter configuration mode, another user is also in configuration mode, a notification message is displayed that indicates who the user is and what portion of the configuration the person is viewing or editing:

```
user@host> configure
Entering configuration mode
```

Copyright © 2014, Juniper Networks, Inc.

```
Users currently editing the configuration:
  root terminal d0 (pid 4137) on since 2008-04-09 23:03:07 PDT, idle 7w6d 08:22
      [edit]
The configuration has been changed but not committed

[edit]
user@host#
```

Related Documentation

- Entering and Exiting the Junos OS CLI Configuration Mode on page 64
- Controlling the Junos OS CLI Environment on page 229

Interface Naming Conventions Used in the Junos OS Operational Commands

This topic explains the interface naming conventions used in the Junos OS operational commands, and contains the following sections:

- Physical Part of an Interface Name on page 158
- Logical Part of an Interface Name on page 158
- Channel Identifier Part of an Interface Name on page 159

Physical Part of an Interface Name

The physical interface naming conventions for Junos OS platforms is as follows:

- On J Series and SRX devices, the unique name of each network interface has the following format to identify the physical device that corresponds to a single physical network connector:

  type-slot/pim-or-ioc/port

- On other platforms, when you display information about an interface, you specify the interface type, the slot in which the Flexible PIC Concentrator (FPC) is installed, the slot on the FPC in which the PIC is located, and the configured port number.

  In the physical part of the interface name, a hyphen (-) separates the media type from the FPC number, and a slash (/) separates the FPC, PIC, and port numbers:

  type-fpc/pic/port

NOTE: Exceptions to the *type-fpc/pic/port* physical description include the aggregated Ethernet and aggregated SONET/SDH interfaces, which use the syntax ae*number* and as*number*, respectively.

Logical Part of an Interface Name

The logical unit part of the interface name corresponds to the logical unit number, which can be a number from 0 through 16,384. In the virtual part of the name, a period (.) separates the port and logical unit numbers:

- J Series and SRX devices:

Copyright © 2014, Juniper Networks, Inc.

type-slot/pim-or-ioc/port:channel.unit

- Other platforms:

  type-fpc/pic/port.logical

Channel Identifier Part of an Interface Name

The channel identifier part of the interface name is required only on channelized interfaces. For channelized interfaces, channel 0 identifies the first channelized interface. For channelized intelligent queuing (IQ) interfaces, channel 1 identifies the first channelized interface.

NOTE: Depending on the type of channelized interface, up to three levels of channelization can be specified. For more information, see the *Junos Network Interfaces Configuration Guide*.

A colon (:) separates the physical and virtual parts of the interface name:

- J Series and SRX devices:

  type-slot/pim-or-ioc/port:channel
  type-slot/pim-or-ioc/port:channel:channel
  type-slot/pim-or-ioc/port:channel:channel:channel

- Other platforms:

  type-fpc/pic/port:channel
  type-fpc//pic/port:channel:channel
  type-fpc/pic/port:channel:channel:channel

Related Documentation

- Example: Configuring Interfaces Using Junos OS Configuration Groups on page 210
- *Junos OS Network Interfaces Library for Routing Devices*

Viewing Files and Directories on a Device Running Junos OS

Junos OS stores information in files on the device, including configuration files, log files, and router software files. This topic shows some examples of operational commands that you can use to view files and directories on a device running Junos OS.

Sections include:

- Directories on the Router or Switch on page 159
- Listing Files and Directories on page 160
- Specifying Filenames and URLs on page 162

Directories on the Router or Switch

Table 9 on page 160 lists some standard directories on a device running Junos OS.

Copyright © 2014, Juniper Networks, Inc.

Table 9: Directories on the Router

Directory	Description
/config	This directory is located on the device's router's internal flash drive. It contains the active configuration (**juniper.conf**) and rollback files 1, 2, and 3.
/var/db/config	This directory is located on the router'sdevice's hard drive and contains rollback files 4 through 49.
/var/tmp	This directory is located on thedevice's hard drive. It holds core files from the various processes on the Routing Engines. Core files are generated when a particular process crashes and are used by Juniper Networks engineers to diagnose the reason for failure.
/var/log	This directory is located on the device's hard drive. It contains files generated by both the device's logging function as well as the **traceoptions** command.
/var/home	This directory is located on the device's hard drive. It contains a subdirectory for each configured user on the device. These individual user directories are the default file location for many Junos OS commands.
/altroot	This directory is located on the device's hard drive and contains a copy of the root file structure from the internal flash drive. This directory is used in certain disaster recovery modes where the internal flash drive is not operational.
/altconfig	This directory is located on the device's hard drive and contains a copy of the **/config** file structure from the internal flash drive. This directory is also used in certain disaster recovery modes when the internal flash drive is not operational.

Listing Files and Directories

You can view the device's directory structure as well as individual files by issuing the **file** command in operational mode.

1. To get help about the **file** command, type the following:

```
user@host> file ?
Possible completions:
  <[Enter]>            Execute this command
  archive              Archives files from the system
  checksum             Calculate file checksum
  compare              Compare files
  copy                 Copy files (local or remote)
  delete               Delete files from the system
  list                 List file information
  rename               Rename files
  show                 Show file contents
  source-address       Local address to use in originating the connection
  |                    Pipe through a command
user@host> file
```


Copyright © 2014, Juniper Networks, Inc.

Help shows that the **file** command includes several options for manipulating files.

2. Use the **list** option to see the directory structure of the device. For example, to show the files located in your home directory on the device:

```
user@host> file list
.ssh/
common
```

The default directory for the **file list** command is the home directory of the user logged in to the device. In fact, the user's home directory is the default directory for most of Junos OS commands requiring a filename.

3. To view the contents of other file directories, specify the directory location. For example:

```
user@host> file list /config
juniper.conf
juniper.conf.1.gz
juniper.conf.2.gz
juniper.conf.3.gz
```

4. You can also use the device's context-sensitive help system to locate a directory. For example:

```
user@host> file list /?
Possible completions:
  <[Enter]>            Execute this command
  <path>               Path to list
  /COPYRIGHT           Size: 6355, Last changed: Feb 13 2005
  /altconfig/          Last changed: Aug 07 2007
  /altroot/            Last changed: Aug 07 2007
  /bin/                Last changed: Apr 09 22:31:35
  /boot/               Last changed: Apr 09 23:28:39
  /config/             Last changed: Apr 16 22:35:35
  /data/               Last changed: Aug 07 2007
  /dev/                Last changed: Apr 09 22:36:21
  /etc/                Last changed: Apr 11 03:14:22
  /kernel              Size: 27823246, Last changed: Aug 07 2007
  /mfs/                Last changed: Apr 09 22:36:49
  /mnt/                Last changed: Jan 11 2007
  /modules/            Last changed: Apr 09 22:33:54
  /opt/                Last changed: Apr 09 22:31:00
  /packages/           Last changed: Apr 09 22:34:38
  /proc/               Last changed: May 07 20:25:46
  /rdm.taf             Size: 498, Last changed: Apr 09 22:37:31
  /root/               Last changed: Apr 10 02:19:45
  /sbin/               Last changed: Apr 09 22:33:55
  /staging/            Last changed: Apr 09 23:28:41
  /tmp/                Last changed: Apr 11 03:14:49
  /usr/                Last changed: Apr 09 22:31:34
  /var/                Last changed: Apr 09 22:37:30
user@host> file list /var/?
<[Enter]>            Execute this command
  <path>               Path to list
  /var/account/        Last changed: Jul 09 2007
  /var/at/             Last changed: Jul 09 2007
  /var/backups/        Last changed: Jul 09 2007
  /var/bin/            Last changed: Jul 09 2007
  /var/crash/          Last changed: Apr 09 22:31:08
  /var/cron/           Last changed: Jul 09 2007
```

Copyright © 2014, Juniper Networks, Inc.

```
  /var/db/             Last changed: May 07 20:28:40
  /var/empty/          Last changed: Jul 09 2007
  /var/etc/            Last changed: Apr 16 22:35:36
  /var/heimdal/        Last changed: Jul 10 2007
  /var/home/           Last changed: Apr 09 22:59:18
  /var/jail/           Last changed: Oct 31 2007
  /var/log/            Last changed: Apr 17 02:00:10
  /var/mail/           Last changed: Jul 09 2007
  /var/msgs/           Last changed: Jul 09 2007
  /var/named/          Last changed: Jul 10 2007
  /var/packages/       Last changed: Jan 18 02:38:59
  /var/pdb/            Last changed: Oct 31 2007
  /var/preserve/       Last changed: Jul 09 2007
  /var/run/            Last changed: Apr 17 02:00:01
  /var/rundb/          Last changed: Apr 17 00:46:00
  /var/rwho/           Last changed: Jul 09 2007
  /var/sdb/            Last changed: Apr 09 22:37:31
  /var/spool/          Last changed: Jul 09 2007
  /var/sw/             Last changed: Jul 09 2007
  /var/tmp/            Last changed: Apr 09 23:28:41
  /var/transfer/       Last changed: Jul 09 2007
  /var/yp/             Last changed: Jul 09 2007
user@host> file list /var/
```

5. You can also display the contents of a file. For example:

```
user@host>file show /var/log/inventory
Jul  9 23:17:46 CHASSISD release 8.4I0 built by builder on 2007-06-12 07:58:27
 UTC
Jul  9 23:18:05 CHASSISD release 8.4I0 built by builder on 2007-06-12 07:58:27
 UTC
Jul  9 23:18:06 Routing Engine 0 - part number 740-003239, serial number
9000016755
Jul  9 23:18:15 Routing Engine 1 - part number 740-003239, serial number
9001018324
Jul  9 23:19:03 SSB 0 - part number 710-001951, serial number AZ8025
Jul  9 23:19:03 SSRAM bank 0 - part number 710-001385, serial number 243071
Jul  9 23:19:03 SSRAM bank 1 - part number 710-001385, serial number 410608
...
```

Specifying Filenames and URLs

In some CLI commands and configuration statements—including **file copy**, **file archive, load, save**, **set system login user** *username* **authentication** *load-key-file*, and **request system software add**—you can include a filename. On a routing matrix, you can include chassis information as part of the filename (for example, **lcc0**, **lcc0-re0**, or **lcc0-re1**).

Copyright © 2014, Juniper Networks, Inc.

You can specify a filename or URL in one of the following ways:

- ***filename***—File in the user's current directory on the local flash drive. You can use wildcards to specify multiple source files or a single destination file. Wildcards are not supported in Hypertext Transfer Protocol (HTTP) or FTP.

  NOTE: **Wildcards are supported only by the file (compare | copy | delete | list | rename | show) commands. When you issue the file show command with a wildcard, it must resolve to one filename.**

- ***path/filename***—File on the local flash disk.
- **/var/*filename*** or **/var/*path/filename***—File on the local hard disk. You can also specify a file on a local Routing Engine for a specific T640 router on a routing matrix:

  ```
  user@host> file delete lcc0-re0:/var/tmp/junk
  ```

- **a:*filename*** or **a:*path/filename***—File on the local drive. The default path is **/** (the root-level directory). The removable media can be in MS-DOS or UNIX (UFS) format.
- ***hostname:/path/filename, hostname:filename, hostname:path/filename,*** or **scp://*hostname/path/filename***—File on an **scp/ssh** client. This form is not available in the worldwide version of Junos OS. The default path is the user's home directory on the remote system. You can also specify ***hostname*** as ***username*@*hostname***.
- **ftp://*hostname/path/filename***—File on an FTP server. You can also specify ***hostname*** as ***username*@*hostname*** or ***username*:*password*@*hostname***. The default path is the user's home directory. To specify an absolute path, the path must start with **%2F**; for example, **ftp://*hostname*/%2F*path/filename***. To have the system prompt you for the password, specify **prompt** in place of the password. If a password is required, and you do not specify the password or **prompt**, an error message is displayed:

  ```
  user@host> file copy ftp://username@ftp.hostname.net//filename
  file copy ftp.hostname.net: Not logged in.

  user@host> file copy ftp://username:prompt@ftp.hostname.net//filename
  Password for username@ftp.hostname.net:
  ```

- **http://*hostname/path/filename***—File on an HTTP server. You can also specify ***hostname*** as ***username*@*hostname*** or ***username*:*password*@*hostname***. If a password is required and you omit it, you are prompted for it.
- **re0:*/path/filename*** or **re1:*/path/filename***—File on a local Routing Engine. You can also specify a file on a local Routing Engine for a specific T640 router on a routing matrix:

  ```
  user@host> show log lcc0-re1:chassisd
  ```

Related Documentation

- Displaying Junos OS Information on page 163

Displaying Junos OS Information

You can display Junos OS version information and other status to determine if the version of Junos OS that you are running supports particular features or hardware.

Copyright © 2014, Juniper Networks, Inc.

To display Junos OS information:

1. Make sure you are in operational mode.
2. To display brief information and status for the kernel and Packet Forwarding Engine, enter the **show version brief** command. This command shows version information for Junos OS packages installed on the router. For example:

```
user@host> show version brief
Hostname: host
Model: m7i
JUNOS Base OS boot [9.1R1.8]
JUNOS Base OS Software Suite [9.1R1.8]
JUNOS Kernel Software Suite [9.1R1.8]
JUNOS Crypto Software Suite [9.1R1.8]
JUNOS Packet Forwarding Engine Support (M/T Common) [9.1R1.8]
JUNOS Packet Forwarding Engine Support (M7i/M10i) [9.1R1.8]
JUNOS Online Documentation [9.1R1.8]
JUNOS Routing Software Suite [9.1R1.8]

user@host>
```

   If the **Junos Crypto Software Suite** is listed, the router has Canada and USA encrypted Junos OS. If the **Junos Crypto Software Suite** is not listed, the router is running worldwide nonencrypted Junos OS.

3. To display detailed version information, enter the **show version detail** command. This command display shows the hostname and version information for Junos OS packages installed on your router. It also includes the version information for each software process. For example:

```
user@host> show version detail

Hostname: host
Model: m20
JUNOS Base OS boot [8.4R1.13]
JUNOS Base OS Software Suite [8.4R1.13]
JUNOS Kernel Software Suite [8.4R1.13]
JUNOS Crypto Software Suite [8.4R1.13]
JUNOS Packet Forwarding Engine Support (M/T Common) [8.4R1.13]
JUNOS Packet Forwarding Engine Support (M20/M40) [8.4R1.13]
JUNOS Online Documentation [8.4R1.13]
JUNOS Routing Software Suite [8.4R1.13]
KERNEL 8.4R1.13 #0 built by builder on 2007-08-08 00:33:41 UTC
MGD release 8.4R1.13 built by builder on 2007-08-08 00:34:00 UTC
CLI release 8.4R1.13 built by builder on 2007-08-08 00:34:47 UTC
RPD release 8.4R1.13 built by builder on 2007-08-08 00:45:21 UTC
CHASSISD release 8.4R1.13 built by builder on 2007-08-08 00:36:59 UTC
DFWD release 8.4R1.13 built by builder on 2007-08-08 00:39:32 UTC
DCD release 8.4R1.13 built by builder on 2007-08-08 00:34:24 UTC
SNMPD release 8.4R1.13 built by builder on 2007-08-08 00:42:24 UTC
MIB2D release 8.4R1.13 built by builder on 2007-08-08 00:46:47 UTC
APSD release 8.4R1.13 built by builder on 2007-08-08 00:36:39 UTC
VRRPD release 8.4R1.13 built by builder on 2007-08-08 00:45:44 UTC
ALARMD release 8.4R1.13 built by builder on 2007-08-08 00:34:30 UTC
PFED release 8.4R1.13 built by builder on 2007-08-08 00:41:54 UTC
CRAFTD release 8.4R1.13 built by builder on 2007-08-08 00:39:03 UTC
SAMPLED release 8.4R1.13 built by builder on 2007-08-08 00:36:05 UTC
ILMID release 8.4R1.13 built by builder on 2007-08-08 00:36:51 UTC
RMOPD release 8.4R1.13 built by builder on 2007-08-08 00:42:04 UTC
```


Copyright © 2014, Juniper Networks, Inc.

```
COSD release 8.4R1.13 built by builder on 2007-08-08 00:38:39 UTC
FSAD release 8.4R1.13 built by builder on 2007-08-08 00:43:01 UTC
IRSD release 8.4R1.13 built by builder on 2007-08-08 00:35:37 UTC
FUD release 8.4R1.13 built by builder on 2007-08-08 00:44:36 UTC
RTSPD release 8.4R1.13 built by builder on 2007-08-08 00:29:14 UTC
SMARTD release 8.4R1.13 built by builder on 2007-08-08 00:13:32 UTC
KSYNCD release 8.4R1.13 built by builder on 2007-08-08 00:33:17 UTC
SPD release 8.4R1.13 built by builder on 2007-08-08 00:43:50 UTC
L2TPD release 8.4R1.13 built by builder on 2007-08-08 00:43:12 UTC
HTTPD release 8.4R1.13 built by builder on 2007-08-08 00:36:27 UTC
PPPOED release 8.4R1.13 built by builder on 2007-08-08 00:36:04 UTC
RDD release 8.4R1.13 built by builder on 2007-08-08 00:33:49 UTC
PPPD release 8.4R1.13 built by builder on 2007-08-08 00:45:13 UTC
DFCD release 8.4R1.13 built by builder on 2007-08-08 00:39:11 UTC
DLSWD release 8.4R1.13 built by builder on 2007-08-08 00:42:37 UTC
LACPD release 8.4R1.13 built by builder on 2007-08-08 00:35:41 UTC
USBD release 8.4R1.13 built by builder on 2007-08-08 00:30:01 UTC
LFMD release 8.4R1.13 built by builder on 2007-08-08 00:35:52 UTC
CFMD release 8.4R1.13 built by builder on 2007-08-08 00:34:45 UTC
JDHCPD release 8.4R1.13 built by builder on 2007-08-08 00:35:40 UTC
PGCPD release 8.4R1.13 built by builder on 2007-08-08 00:46:31 UTC
SSD release 8.4R1.13 built by builder on 2007-08-08 00:36:17 UTC
MSPD release 8.4R1.13 built by builder on 2007-08-08 00:33:42 UTC
KMD release 8.4R1.13 built by builder on 2007-08-08 00:44:02 UTC
PPMD release 8.4R1.13 built by builder on 2007-08-08 00:36:03 UTC
LMPD release 8.4R1.13 built by builder on 2007-08-08 00:33:49 UTC
LRMUXD release 8.4R1.13 built by builder on 2007-08-08 00:33:55 UTC
PGMD release 8.4R1.13 built by builder on 2007-08-08 00:36:01 UTC
BFDD release 8.4R1.13 built by builder on 2007-08-08 00:44:22 UTC
SDXD release 8.4R1.13 built by builder on 2007-08-08 00:36:18 UTC
AUDITD release 8.4R1.13 built by builder on 2007-08-08 00:34:40 UTC
L2ALD release 8.4R1.13 built by builder on 2007-08-08 00:40:05 UTC
EVENTD release 8.4R1.13 built by builder on 2007-08-08 00:39:55 UTC
L2CPD release 8.4R1.13 built by builder on 2007-08-08 00:41:04 UTC
MPLSOAMD release 8.4R1.13 built by builder on 2007-08-08 00:45:11 UTC
jroute-dd release 8.4R1.13 built by builder on 2007-08-08 00:31:01 UTC
jkernel-dd release 8.4R1.13 built by builder on 2007-08-08 00:30:30 UTC
jcrypto-dd release 8.4R1.13 built by builder on 2007-08-08 00:30:12 UTC
jdocs-dd release 8.4R1.13 built by builder on 2007-08-08 00:02:52 UTC

user@host>
```

Related Documentation

- Managing Programs and Processes Using Junos OS Operational Mode Commands on page 165

Managing Programs and Processes Using Junos OS Operational Mode Commands

This topic shows some examples of Junos operational commands that you can use to manage programs and processes on a device running Junos OS.

Sections include:

- Showing Software Processes on page 166
- Restarting a Junos OS Process on page 167
- Stopping Junos OS on page 168
- Rebooting Junos OS on page 169

Copyright © 2014, Juniper Networks, Inc.

Showing Software Processes

To verify system operation or to begin diagnosing an error condition, you may need to display information about software processes running on the device.

To show software processes:

1. Make sure you are in operational mode.
2. Type the **show system processes extensive** command. This command shows the CPU utilization on the device and lists the processes in order of CPU utilization. For example:

```
user@host> show system processes extensive

last pid: 28689;  load averages:  0.01,  0.00,  0.00  up 56+06:16:13    04:52:04
73 processes:  1 running, 72 sleeping

Mem: 101M Active, 101M Inact, 98M Wired, 159M Cache, 69M Buf, 286M Free
Swap: 1536M Total, 1536M Free


  PID USERNAME    PRI NICE  SIZE    RES STATE    TIME   WCPU    CPU COMMAND
 3365 root          2   0 21408K  4464K select 511:23  0.00%  0.00% chassisd
 3508 root          2   0  3352K  1168K select  32:45  0.00%  0.00% l2ald
 3525 root          2   0  3904K  1620K select  13:40  0.00%  0.00% dcd
 5532 root          2   0 11660K  2856K kqread  10:36  0.00%  0.00% rpd
 3366 root          2   0  2080K   828K select   8:33  0.00%  0.00% alarmd
 3529 root          2   0  2040K   428K select   7:32  0.00%  0.00% irsd
 3375 root          2   0  2900K  1600K select   6:01  0.00%  0.00% ppmd
 3506 root          2   0  5176K  2568K select   5:38  0.00%  0.00% mib2d
 4957 root          2   0  1284K   624K select   5:16  0.00%  0.00% ntpd
    6 root         18   0     0K     0K syncer   4:49  0.00%  0.00% syncer
 3521 root          2   0  2312K   928K select   2:14  0.00%  0.00% lfmd
 3526 root          2   0  5192K  1988K select   2:04  0.00%  0.00% snmpd
 3543 root          2   0     0K     0K peer_s   1:46  0.00%  0.00% peer proxy

 3512 root          2   0  3472K  1044K select   1:44  0.00%  0.00% rmopd
 3537 root          2   0     0K     0K peer_s   1:30  0.00%  0.00% peer proxy

 3527 root          2   0  3100K  1176K select   1:14  0.00%  0.00% pfed
 3380 root          2   0  3208K  1052K select   1:11  0.00%  0.00% bfdd
 4136 root          2   0 11252K  3668K select   0:54  0.00%  0.00% cli
 3280 root          2   0  2248K  1420K select   0:28  0.00%  0.00% eventd
 3528 root          2   0  2708K   672K select   0:28  0.00%  0.00% dfwd
    7 root         -2   0     0K     0K vlruwt   0:26  0.00%  0.00% vnlru
 3371 root          2   0  1024K   216K sbwait   0:25  0.00%  0.00% tnp.sntpd

   13 root        -18   0     0K     0K psleep   0:24  0.00%  0.00% vmuncacheda

 3376 root          2   0  1228K   672K select   0:22  0.00%  0.00% smartd
    5 root        -18   0     0K     0K psleep   0:17  0.00%  0.00% bufdaemon

 3368 root          2   0 15648K  9428K select   0:17  0.00%  0.00% mgd
 3362 root          2   0  1020K   204K select   0:15  0.00%  0.00% watchdog
 3381 root          2   0  2124K   808K select   0:15  0.00%  0.00% lacpd
 3524 root          2   0  6276K  1492K select   0:14  0.00%  0.00% kmd
 3343 root         10   0  1156K   404K nanslp   0:14  0.00%  0.00% cron
---(more)---
```


Copyright © 2014, Juniper Networks, Inc.

Table 10 on page 167 lists and describes the output fields included in this example. The fields are listed in alphabetical order.

Table 10: show system process extensive Command Output Fields

Field	Description
COMMAND	Command that is running.
CPU	Raw (unweighted) CPU usage. The value of this field is used to sort the processes in the output.
last pid	Last process identifier assigned to the process.
load averages	Three load averages, followed by the current time.
Mem	Information about physical and virtual memory allocation.
NICE	UNIX "nice" value. The nice value allows a process to change its final scheduling priority.
PID	Process identifier.
PRI	Current kernel scheduling priority of the process. A lower number indicates a higher priority.
processes	Number of existing processes and the number of processes in each state (**sleeping, running, starting, zombies,** and **stopped**).
RES	Current amount of resident memory, in KB.
SIZE	Total size of the process (**text, data,** and **stack**), in KB.
STATE	Current state of the process (**sleep**, **wait**, **run**, **idle**, **zombi**, or **stop**).
Swap	Information about physical and virtual memory allocation.
USERNAME	Owner of the process.
WCPU	Weighted CPU usage.

Restarting a Junos OS Process

To correct an error condition, you might need to restart a software process running on the device. You can use the **restart** command to force a restart of a software process.

CAUTION: Do not restart a software process unless specifically asked to do so by your Juniper Networks customer support representative. Restarting a software process during normal operation of a device could cause interruption of packet forwarding and loss of data.

Copyright © 2014, Juniper Networks, Inc.

To restart a software process:

1. Make sure you are in operational mode.
2. Type the following command:

   user@host> **restart** ***process-name*** **< (immediately | gracefully | soft) >**

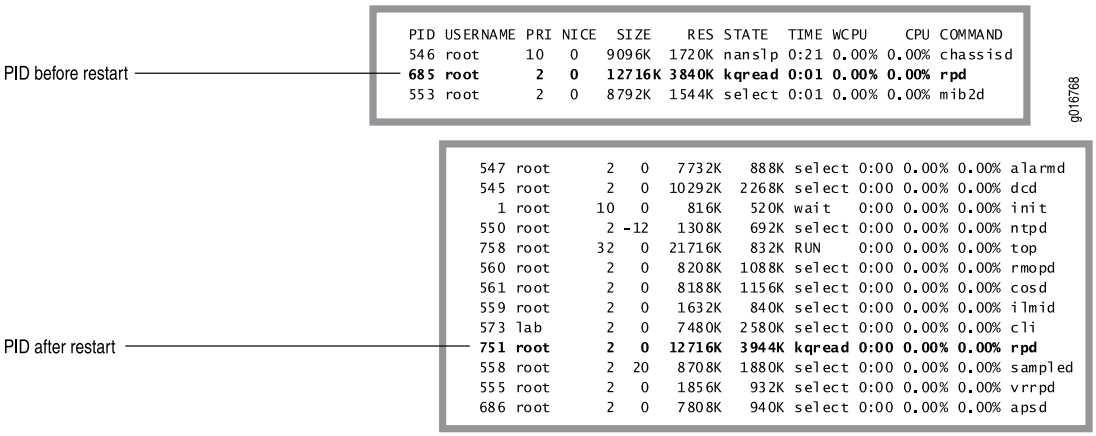
   - ***process-name*** is the name of the process that you want to restart. For example, **routing** or **class-of-service**. You can use the command completion feature of Junos OS to see a list of software processes that you can restart using this command.
   - **gracefully** restarts the software process after performing clean-up tasks.
   - **immediately** restarts the software process without performing any clean-up tasks.
   - **soft** rereads and reactivates the configuration without completely restarting the software processes. For example, BGP peers stay up and the routing table stays constant.

The following example shows how to restart the routing process:

```
user@host> restart routing
Routing protocol daemon started, pid 751
```

When a process restarts, the process identifer (PID) is updated. (See Figure 13 on page 168.)

Figure 13: Restarting a Process



Stopping Junos OS

To avoid damage to the file system and to prevent loss of data, you must always gracefully shut down Junos OS before powering off the device.

Copyright © 2014, Juniper Networks, Inc.

NOTE: SRX Series Services Gateway devices for the branch and EX Series Ethernet Switches support resilient dual-root partitioning.

If you are unable to shut down a device gracefully because of unexpected circumstances such as a power outage or a device failure, resilient dual-root partitioning prevents file corruption and enables a device to remain operational. In addition, it enables a device to boot transparently from the second root partition if the system fails to boot from the primary root partition.

Resilient dual-root partitioning serves as a backup mechanism for providing additional resiliency to a device when there is an abnormal shutdown. However, it is not an alternative to performing a graceful shutdown under normal circumstances.

To stop Junos OS:

1. Make sure you are in operational mode.
2. Enter the **request system halt** command. This command stops all system processes and halts the operating system. For example:

```
user@host> request system halt
Halt the system? [yes,no] (no)  yes
shutdown: [pid 3110]
Shutdown NOW!
*** FINAL System shutdown message from root@host ***
System going down IMMEDIATELY
user@host> Dec 17 17:28:40 init: syslogd (PID 2514) exited with status=0 Normal
 Exit
Waiting (max 60 seconds) for system process `bufdaemon' to stop...stopped
Waiting (max 60 seconds) for system process `syncer' to stop...stopped
syncing disks... 4
done
Uptime: 3h31m41s
ata0: resetting devices.. done
The operating system has halted.
Please press any key to reboot.
```

Rebooting Junos OS

After a software upgrade or to recover (occasionally) from an error condition, you must reboot Junos OS.

To reboot the Junos OS:

1. Make sure you are in operational mode.
2. Enter the **request system reboot** command. This command displays the final stages of the system shutdown and executes the reboot. Reboot requests are recorded to the system log files, which you can view with the **show log messages** command. For example:

```
user@host>request system rebootReboot the system? [yes,no] (no)yes
```

Copyright © 2014, Juniper Networks, Inc.

```
shutdown: [pid 845]
Shutdown NOW!
*** FINAL System shutdown message from root@host ***
System going down IMMEDIATELY
user@host> Dec 17 17:34:20 init: syslogd (PID 409) exited with status=0 Normal
 Exit
Waiting (max 60 seconds) for system process `bufdaemon' to stop...stopped
Waiting (max 60 seconds) for system process `syncer' to stop...stopped
syncing disks... 10 6
done
Uptime: 2m45s
ata0: resetting devices.. done
Rebooting...
```

Related Documentation

- Checking the Status of a Device Running Junos OS on page 38
- Displaying Junos OS Information on page 163
- *Understanding Resilient Dual-Root Partitions on Switches*

Using the Junos OS CLI Comment Character # for Operational Mode Commands

The comment character in Junos OS enables you to copy operational mode commands that include comments from a file and paste them into the CLI. A pound sign (#) at the beginning of the command-line indicates a comment line. This is useful for describing frequently used operational mode commands; for example, a user's work instructions on how to monitor the network. To add a comment to a command file, the first character of the line must be #. When you start a command with #, the rest of the line is disregarded by Junos OS.

To add comments in operational mode, start with a # and end with a new line (carriage return):

```
user@host> # comment-string
```

comment-string is the text of the comment. The comment text can be any length, but each comment line must begin with a #.

Related Documentation

- Example: Using Comments in Junos OS Operational Mode Commands on page 170

Example: Using Comments in Junos OS Operational Mode Commands

The following example shows how to use comments in a file:

```
#Command 1: Show the router version
show version
#Command 2: Show all router interfaces
show interfaces terse
```

The following example shows how to copy and paste contents of a file into the CLI:

```
user@host> #Command 1: Show the router version
user@host> show version
Hostname: myhost
```


Copyright © 2014, Juniper Networks, Inc.

```
Model: m5
Junos Base OS boot [6.4-20040511.0]
Junos Base OS Software Suite [6.4-20040511.0]
Junos Kernel Software Suite [6.4-20040511.0]
Junos Packet Forwarding Engine Support (M5/M10) [6.4-20040511.0] Junos Routing
  Software Suite [6.4-20040511.0] Junos Online Documentation [6.4-20040511.0] Junos
  Crypto Software Suite [6.4-20040511.0]
user@host> # Command 2: Show all router interfaces
user@host> show interfaces terse
Interface Admin Link Proto Local Remote
fe-0/0/0 up up
fe-0/0/1 up down
fe-0/0/2 up down
mo-0/1/0 up
mo-0/1/0.16383 up up inet 10.0.0.1 --> 10.0.0.17
so-0/2/0 up up
so-0/2/1 up up
dsc up up
fxp0 up up
fxp0.0 up up inet 192.168.70.62/21
fxp1 up up
fxp1.0 up up tnp 4
gre up up
ipip up up
lo0 up up
lo0.0 up up inet 127.0.0.1 --> 0/0
lo0.16385 up up inet
```

Related Documentation

- Using the Junos OS CLI Comment Character # for Operational Mode Commands on page 170

Copyright © 2014, Juniper Networks, Inc.

Copyright © 2014, Juniper Networks, Inc.

CHAPTER 8

Filtering Command Output

- Using the Pipe (|) Symbol to Filter Junos OS Command Output on page 173
- Using Regular Expressions with the Pipe (|) Symbol to Filter Junos OS Command Output on page 174
- Filtering Operational Mode Command Output in a QFabric System on page 175
- Pipe (|) Filter Functions in the Junos OS Command-Line Interface on page 176

Using the Pipe (|) Symbol to Filter Junos OS Command Output

The Junos OS enables you to filter command output by adding the pipe (**|**) symbol when you enter a command.

For example:

```
user@host> show rip neighbor ?
Possible completions:
  <[Enter]>            Execute this command
  <name>               Name of RIP neighbor
  instance             Name of RIP instance
  logical-system       Name of logical system, or 'all'
  |                    Pipe through a command
```

The following example lists the filters that can be used with the pipe symbol (|):

```
user@host> show interfaces | ?
astatti@allspice> show interfaces | ?
Possible completions:
  append               Append output text to file
  count                Count occurrences
  display              Show additional kinds of information
  except               Show only text that does not match a pattern
  find                 Search for first occurrence of pattern
  hold                 Hold text without exiting the --More-- prompt
  last                 Display end of output only
  match                Show only text that matches a pattern
  no-more              Don't paginate output
  refresh              Refresh a continuous display of the command
  request              Make system-level requests
  resolve              Resolve IP addresses
  save                 Save output text to file
  tee                  Write to standard output and file
  trim                 Trim specified number of columns from start of line
```

Copyright © 2014, Juniper Networks, Inc.

For the **show configuration** command only, an additional compare filter is available:

```
user@host> show configuration | ?
Possible completions:
  compare             Compare configuration changes with prior version
...
```

You can enter any of the pipe filters in conjunction. For example:

user@host>*command* **|** match *regular-expression* **|** save *filename*

NOTE: This topic describes *only* the filters that can be used for operational mode command output. For information about filters that can be used in configuration mode, see the *Junos OS Administration Library for Routing Devices*.

Related Documentation

- Pipe (|) Filter Functions in the Junos OS Command-Line Interface on page 176
- Using Regular Expressions with the Pipe (|) Symbol to Filter Junos OS Command Output on page 174
- Filtering Operational Mode Command Output in a QFabric System on page 175

Using Regular Expressions with the Pipe (|) Symbol to Filter Junos OS Command Output

The **except**, **find**, and **match** filters used with the pipe symbol employ regular expressions to filter output. Juniper Networks uses the regular expressions as defined in POSIX 1003.2. If the regular expressions contain spaces, operators, or wildcard characters, enclose the expression in quotation marks.

Table 11: Common Regular Expression Operators in Operational Mode Commands

Operator	Function
|	Indicates that a match can be one of the two terms on either side of the pipe.
^	Used at the beginning of an expression, denotes where a match should begin.
$	Used at the end of an expression, denotes that a term must be matched exactly up to the point of the **$** character.
[ ]	Specifies a range of letters or digits to match. To separate the start and end of a range, use a hyphen (-).
( )	Specifies a group of terms to match.

For example, if a command produces the following output:

Copyright © 2014, Juniper Networks, Inc.

```
12
22
321
4
```

a pipe filter of **| match 2** displays the following output:

```
12
22
321
```

and a pipe filter of **| except 1** displays the following output:

```
22
4
```

Related Documentation

- Using the Pipe (|) Symbol to Filter Junos OS Command Output on page 173
- Pipe (|) Filter Functions in the Junos OS Command-Line Interface on page 176

Filtering Operational Mode Command Output in a QFabric System

When you issue an operational mode command in a QFabric system, the output generated can be fairly extensive because of the number of components contained within the system. To make the output more accessible, you can filter the output by appending the **| filter** option to the end of most Junos OS commands.

1. To filter operational mode command output and limit it to a Node group, include the **| filter node-group** *node-group-name* option at the end of your Junos OS operational mode command.

   root@qfabric> show interfaces terse | filter node-group NW-NG-0

   ```
   Interface               Admin Link Proto    Local                 Remote
   NW-NG-0:dsc             up    up
   NW-NG-0:em0             up    up
   NW-NG-0:em1             up    up
   NW-NG-0:gre             up    up
   NW-NG-0:ipip            up    up
   NW-NG-0:lo0             up    up
   NW-NG-0:lo0.16384       up    up   inet     127.0.0.1           --> 0/0
   NW-NG-0:lo0.16385       up    up   inet
   NW-NG-0:lsi             up    up
   NW-NG-0:mtun            up    up
   NW-NG-0:pimd            up    up
   NW-NG-0:pime            up    up
   NW-NG-0:tap             up    up
   Node01:ge-0/0/10        up    up
   Node01:ge-0/0/40        up    up
   Node01:ge-0/0/41        up    up
   vlan                    up    up
   ```

2. To filter operational mode command output and limit it to a set of Node groups, include the **| filter node-group** option at the end of your Junos OS operational mode command and specify the list of Node group names in brackets.

   root@qfabric> show ethernet-switching interfaces | filter node-group [NW-NG-0 RSNG-1]

Copyright © 2014, Juniper Networks, Inc.

```
Interface     State  VLAN members        Tag   Tagging  Blocking
NW-NG-0:ae0.0 up     v200                200   tagged   unblocked
                     v50                 50    tagged   unblocked
                     v51                 51    tagged   unblocked
                     v52                 52    tagged   unblocked
                     v53                 53    tagged   unblocked
RSNG-1:ae0.0  up     v200                200   untagged unblocked
RSNG-1:ae47.0 up     v50                 50    tagged   unblocked
                     v51                 51    tagged   unblocked
                     v52                 52    tagged   unblocked
                     v53                 53    tagged   unblocked
```

Related Documentation

- *QFabric System Operational Mode Commands*
- Using the Pipe (|) Symbol to Filter Junos OS Command Output on page 173

Pipe (|) Filter Functions in the Junos OS Command-Line Interface

This topic describes the pipe (|) filter functions that are supported in the Junos OS command-line interface (CLI):

- Comparing Configurations on page 176
- Counting the Number of Lines of Output on page 178
- Displaying Output in XML Tag Format on page 178
- Displaying the RPC tags for a Command on page 178
- Ignoring Output That Does Not Match a Regular Expression on page 179
- Displaying Output from the First Match of a Regular Expression on page 179
- Retaining Output After the Last Screen on page 179
- Displaying Output Beginning with the Last Entries on page 180
- Displaying Output That Matches a Regular Expression on page 180
- Preventing Output from Being Paginated on page 180
- Sending Command Output to Other Users on page 181
- Resolving IP Addresses on page 181
- Saving Output to a File on page 181
- Appending Output to a File on page 182
- Displaying Output on Screen and Writing to a File on page 182
- Trimming Output by Specifying the Starting Column on page 182
- Refreshing the Output of a Command on page 183

Comparing Configurations

The **compare** filter compares the candidate configuration with either the current committed configuration or a configuration file and displays the differences between the two configurations. To compare configurations, enter **compare** after the pipe (|) symbol:

```
[edit]
```


Copyright © 2014, Juniper Networks, Inc.

user@host# **show | compare** [*filename*| **rollback** *n*]

filename is the full path to a configuration file.

n is the index into the list of previously committed configurations. The most recently saved configuration is 0. If you do not specify arguments, the candidate configuration is compared against the active configuration file (**/config/juniper.conf**).

The comparison output uses the following conventions:

- Statements that are only in the candidate configuration are prefixed with a plus sign (+).
- Statements that are only in the comparison file are prefixed with a minus sign (–).
- Statements that are unchanged are prefixed with a single blank space ( ).

For example:

```
user@host> show configuration system | compare rollback 9
[edit system]
+ host-name nutmeg;
+ backup-router 192.168.71.254;
- ports {
-     console log-out-on-disconnect;
- }
[edit system name-server]
+  172.17.28.11;
   172.17.28.101 { ... }
[edit system name-server]
   172.17.28.101 { ... }
+  172.17.28.100;
+  172.17.28.10;
[edit system]
-  scripts {
-      commit {
-          allow-transients;
-      }
-  }
+  services {
+      ftp;
+      rlogin;
+      rsh;
+      telnet;
+  }
```

Starting with Junos OS Release 8.3, output from the **show | compare** command has been enhanced to more accurately reflect configuration changes. This includes more intelligent handling of order changes in lists. For example, consider names in a group that are reordered as follows:

```
groups {     groups {
group_xmp;      group_xmp;
group_cmp;     group_grp:
group_grp;     group_cmp;
}     }
```

Copyright © 2014, Juniper Networks, Inc.

In previous releases, output from the **show | compare** command looked like the following:

```
[edit groups]
  - group_xmp;
  - group_cmp;
  - group_grp;
  + group_xmp;
  + group_grp;
  + group_cmp;
```

Now, output from the **show | compare** command looks like the following:

```
[edit groups]
group_xmp {...}
!  group_grp {...}
```

Counting the Number of Lines of Output

To count the number of lines in the output from a command, enter **count** after the pipe symbol (|). For example:

```
user@host> show configuration | count
Count: 269 lines
```

Displaying Output in XML Tag Format

To display command output in XML tag format, enter **display xml** after the pipe symbol (|).

The following example displays the **show cli directory** command output as XML tags:

```
user@host> show cli directory | display xml
<rpc-reply xmlns:junos="http://xml.juniper.net/junos/7.5I0/junos">
    <cli>
        <working-directory>/var/home/regress</working-directory>
    </cli>
    <cli>
        <banner></banner>
    </cli>
</rpc-reply>
```

Displaying the RPC tags for a Command

To display the remote procedure call (RPC) XML tags for an operational mode command, enter **display xml rpc** after the pipe symbol (|).

The following example displays the RPC tags for the **show route** command:

```
user@host> show route | display xml rpc
<rpc-reply xmlns:junos="http://xml.juniper.net/junos/10.1I0/junos">
    <rpc>
        <get-route-information>
        </get-route-information>
    </rpc>
    <cli>
        <banner></banner>
    </cli>
</rpc-reply>
```


Copyright © 2014, Juniper Networks, Inc.

Ignoring Output That Does Not Match a Regular Expression

To ignore text that matches a regular expression, specify the **except** command after the pipe symbol (|). If the regular expression contains any spaces, operators, or wildcard characters, enclose it in quotation marks. For information on common regular expression operators, see "Using Regular Expressions with the Pipe (|) Symbol to Filter Junos OS Command Output" on page 174.

The following example displays all users who are logged in to the router, except for the user **root**:

```
user@host> show system users | except root
 8:28PM  up 1 day, 13:59, 2 users, load averages: 0.01, 0.01, 0.00
USER     TTY FROM              LOGIN@  IDLE WHAT
sheep    p0  baa.juniper.net   7:25PM     - cli
```

Displaying Output from the First Match of a Regular Expression

To display output starting with the first occurrence of text matching a regular expression, enter **find** after the pipe symbol (|). If the regular expression contains any spaces, operators, or wildcard characters, enclose it in quotation marks. For information on common regular expression operators, see "Using Regular Expressions with the Pipe (|) Symbol to Filter Junos OS Command Output" on page 174.

The following example displays the routes in the routing table starting at IP address **208.197.169.0**:

```
user@host> show route | find 208.197.169.0
208.197.169.0/24   *[Static/5] 1d 13:22:11
                    > to 192.168.4.254 via so-3/0/0.0
224.0.0.5/32       *[OSPF/10] 1d 13:22:12, metric 1
iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
47.0005.80ff.f800.0000.0108.0001.1921.6800.4015.00/160
                   *[Direct/0] 1d 13:22:12
                    > via lo0.0
```

The following example displays the first CCC entry in the forwarding table:

```
user@host> show route forwarding-table | find ccc
Routing table: ccc
MPLS:
Interface.Label    Type RtRef Nexthop       Type Index NhRef Netif
default            perm     0               rjct     3     1
0                  user     0               recv     5     2
1                  user     0               recv     5     2
32769              user     0               ucst    45     1 fe-0/0/0.534
fe-0/0/0. (CCC)    user     0               indr    44     2
                              10.0.16.2     Push 32768, Push
```

Retaining Output After the Last Screen

To not return immediately to the CLI prompt after viewing the last screen of output, enter **hold** after the pipe symbol (|). The following example prevents returning to the CLI

Copyright © 2014, Juniper Networks, Inc.

prompt after you have viewed the last screen of output from the **show log log-file-1** command:

```
user@host> show log log-file-1 | hold
```

This filter is useful when you want to scroll or search through output.

Displaying Output Beginning with the Last Entries

To display text starting from the end of the output, enter **last <*lines*>** after the pipe symbol (|).

The following example displays the last entries in **log-file-1** file:

```
user@host> show log log-file-1 | last
```

This filter is useful for viewing log files in which the end of the file contains the most recent entries.

NOTE: When the number of lines requested is less than the number of lines that the screen length setting permits you to display, Junos returns as many lines as permitted by the screen length setting. That is, if your screen length is set to 20 lines and you have requested only the last 10 lines, Junos returns the last 19 lines instead of the last 10 lines.

Displaying Output That Matches a Regular Expression

To display output that matches a regular expression, enter **match *regular-expression*** after the pipe symbol (|). If the regular expression contains any spaces, operators, or wildcard characters, enclose it in quotation marks. For information on common regular expression operators, see "Using Regular Expressions with the Pipe (|) Symbol to Filter Junos OS Command Output" on page 174.

The following example matches all the Asynchronous Transfer Mode (ATM) interfaces in the configuration:

```
user@host> show configuration | match at-
at-2/1/0 {
at-2/1/1 {
at-2/2/0 {
at-5/2/0 {
at-5/3/0 {
```

Preventing Output from Being Paginated

By default, if output is longer than the length of the terminal screen, you are provided with a **---(more)---** message to display the remaining output. To display the remaining output, press the Spacebar.

To prevent the output from being paginated, enter **no-more** after the pipe symbol (|).

The following example displays output from the **show configuration** command all at once:

```
user@host> show configuration | no-more
```


Copyright © 2014, Juniper Networks, Inc.

This feature is useful, for example, if you want to copy the entire output and paste it into an e-mail.

Sending Command Output to Other Users

To display command output on the terminal of a specific user logged in to your router, or on the terminals of all users logged in to your router, enter **request message (all | user *account*@*terminal*)** after the pipe symbol (|).

If you are troubleshooting your router and, for example, talking with a customer service representative on the phone, you can use the **request message** command to send your representative the command output you are currently viewing on your terminal.

The following example sends the output from the **show interfaces** command you enter on your terminal to the terminal of the user **root@ttyp1**:

```
user@host> show interfaces | request message user root@ttyp1
```

The user **root@ttyp1** sees the following output appear on the terminal screen:

```
Message from user@host on /dev/ttyp0 at 10:32 PST...
Physical interface: dsc, Enabled, Physical link is Up
   Interface index: 5, SNMP ifIndex: 5
   Type: Software-Pseudo, MTU: Unlimited...
```

Resolving IP Addresses

In operational mode only, if the output of a command displays an unresolved IP address, you can enter **| resolve** after the command to display the name associated with the IP address. The **resolve** filter enables the system to perform a reverse DNS lookup of the IP address. If DNS is not enabled, the lookup fails and no substitution is performed.

To perform a reverse DNS lookup of an unresolved IP address, enter **resolve <full-names>** after the pipe symbol (|). If you do not specify the **full-names** option, the name is truncated to fit whatever field width limitations apply to the IP address.

The following example performs a DNS lookup on any unresolved IP addresses in the output from the **show ospf neighbors** command:

```
user@host> show ospf neighbors | resolve
```

Saving Output to a File

When command output is lengthy, when you need to store or analyze the output, or when you need to send the output in an e-mail or by FTP, you can save the output to a file. By default, the file is placed in your home directory on the router.

To save command output to a file, enter **save *filename*** after the pipe symbol (|).

The following example saves the output from the **request support information** command to a file named **my-support-info.txt:**

```
user@host> request support information | save my-support-info.txt
Wrote 1143 lines of output to 'my-support-info.txt'
user@host>
```

Copyright © 2014, Juniper Networks, Inc.

Appending Output to a File

When command output is displayed, you can either save the output to a file, which overwrites the existing contents of that file or you can append the output text to a specific file.

To append the command output to the file, enter **append *filename*** after the pipe symbol (|).

The following example appends the output from the **request support information** command to a file named **my-support-info.txt:**

```
user@host> request support information | append my-support-info.txt
Wrote 2247 lines of output to 'my-support-info.txt'
user@host>
```

Displaying Output on Screen and Writing to a File

When command output is displayed, you can also write the output to a file. To both display the output and write it to a file, enter **tee *filename*** after the pipe symbol (|).

The following example displays the output from the **show interfaces ge-* terse** command (displaying information about the status of the gigabit Ethernet interfaces on the device) and diverts the output to a file called **ge-interfaces.txt**:

```
user@host> show interfaces ge-* terse | tee ge-interfaces.txt
Interface               Admin Link Proto    Local                 Remote
ge-0/1/0                up    down
ge-0/1/1                up    up
ge-0/1/2                up    down
ge-0/1/3                up    up
```

Unlike the UNIX **tee** command, only an error message is displayed if the file cannot be opened (instead of displaying the output and then the error message).

```
user@host> show interfaces ge-* terse | tee /homes/user/test.txt
error: tee failed: file /homes/user/test.txt could not be opened

user@host>
```

Trimming Output by Specifying the Starting Column

Output appears on the terminal screen in terms of rows and columns. The first alphanumeric character starting at the left of the screen is in column 1, the second character is in column 2, and so on. To display output starting from a specific column (thus trimming the leftmost portion of the output), enter **trim columns** after the pipe symbol (|). The **trim** filter is useful for trimming the date and time from the beginning of system log messages

The following example displays output from the **show system storage** command, filtering out the first 10 columns:

```
user@host> show system storage | trim 11
```


Copyright © 2014, Juniper Networks, Inc.

NOTE: **trim** command does not accept negative values.

Refreshing the Output of a Command

You can run an operational mode command with the **| refresh** pipe option to refresh the output displayed on the screen periodically. The default refresh occurs every second. However, you can also explicitly specify a refresh interval from **1** through **604800** seconds. For example, to refresh the output of the **show interfaces** command every five seconds, you would run the following command:

```
user@host > show interfaces | refresh 5
```

Related Documentation

- Using Regular Expressions with the Pipe (|) Symbol to Filter Junos OS Command Output on page 174
- Using the Pipe (|) Symbol to Filter Junos OS Command Output on page 173

Copyright © 2014, Juniper Networks, Inc.

Copyright © 2014, Juniper Networks, Inc.

CHAPTER 9

Using Shortcuts, Wildcards, and Regular Expressions in the CLI

- Using Keyboard Sequences to Move Around and Edit the Junos OS CLI on page 185
- Using Wildcard Characters in Interface Names on page 187
- Common Regular Expressions to Use with the replace Command on page 188
- Using Global Replace in a Junos OS Configuration on page 189
- Example: Using Global Replace in a Junos OS Configuration—Using the \n Back Reference on page 190
- Example: Using Global Replace in a Junos OS Configuration—Replacing an Interface Name on page 191
- Example: Using Global Replace in a Junos OS Configuration—Using the upto Option on page 192
- Using Regular Expressions to Delete Related Items from a Junos OS cConfiguration on page 193

Using Keyboard Sequences to Move Around and Edit the Junos OS CLI

You can use keyboard sequences in the Junos OS command-line interface (CLI) to move around and edit the command line. You can also use keyboard sequences to scroll through a list of recently executed commands. Table 12 on page 186 lists some of the CLI keyboard sequences. They are the same as those used in Emacs.

Copyright © 2014, Juniper Networks, Inc.

Table 12: CLI Keyboard Sequences

Category	Action	Keyboard Sequence
Move the Cursor	Move the cursor back one character.	Ctrl+b
	Move the cursor back one word.	Esc+b or Alt+b
	Move the cursor forward one character.	Ctrl+f
	Move the cursor forward one word.	Esc+f or Alt+f
	Move the cursor to the beginning of the command line.	Ctrl+a
	Move the cursor to the end of the command line.	Ctrl+e
Delete Characters	Delete the character before the cursor.	Ctrl+h, Delete, or Backspace
	Delete the character at the cursor.	Ctrl+d
	Delete all characters from the cursor to the end of the command line.	Ctrl+k
	Delete all characters on the command line.	Ctrl+u or Ctrl+x
	Delete the word before the cursor.	Ctrl+w, Esc+Backspace, or Alt+Backspace
	Delete the word after the cursor.	Esc+d or Alt+d
Insert Recently Deleted Text	Insert the most recently deleted text at the cursor.	Ctrl+y
Redraw the Screen	Redraw the current line.	Ctrl+l

Copyright © 2014, Juniper Networks, Inc.

Table 12: CLI Keyboard Sequences *(continued)*

Category	Action	Keyboard Sequence
Display Previous Command Lines	Scroll backward through the list of recently executed commands.	Ctrl+p
	Scroll forward through the list of recently executed commands.	Ctrl+n
	Search the CLI history in reverse order for lines matching the search string.	Ctrl+r
	Search the CLI history by typing some text at the prompt, followed by the keyboard sequence. The CLI attempts to expand the text into the most recent word in the history for which the text is a prefix.	Esc+/
Display Previous Command Words	Scroll backward through the list of recently entered words in a command line.	Esc+. or Alt+.
Repeat Keyboard Sequences	Specify the number of times to execute a keyboard sequence. *number* can be from 1 through 9 and *sequence* is the keyboard sequence that you want to execute.	Esc+*number sequence* or Alt+*number sequence*

Related Documentation

- Using Wildcard Characters in Interface Names on page 187
- Using Global Replace in a Junos OS Configuration on page 189

Using Wildcard Characters in Interface Names

You can use wildcard characters in the Junos OS operational commands to specify groups of interface names without having to type each name individually. Table 13 on page 187 lists the available wildcard characters. You must enclose all wildcard characters except the asterisk (*) in quotation marks (“ ”).

Table 13: Wildcard Characters for Specifying Interface Names

Wildcard Character	Description
*** (asterisk)**	Match any string of characters in that position in the interface name. For example, **so*** matches all SONET/SDH interfaces.
"[*character*<*character*...>]"	Match one or more individual characters in that position in the interface name. For example, **so-“[03]”*** matches all SONET/SDH interfaces in slots 0 and 3.

Copyright © 2014, Juniper Networks, Inc.

Table 13: Wildcard Characters for Specifying Interface Names *(continued)*

Wildcard Character	Description
"[!*character*<*character*...>]"	Match all characters except the ones included in the brackets. For example, **so-"[!03]"*** matches all SONET/SDH interfaces except those in slots 0 and 3.
"[*character1-character2*]"	Match a range of characters. For example, **so-"[0-3]" *** matches all SONET/SDH interfaces in slots 0, 1, 2, and 3.
"[!*character1-character2*]"	Match all characters that are not in the specified range of characters. For example, **so-"[!0-3]"*** matches all SONET/SDH interfaces in slots 4, 5, 6, and 7.

Related Documentation

- Using Keyboard Sequences to Move Around and Edit the Junos OS CLI on page 185
- Using Global Replace in a Junos OS Configuration on page 189

Common Regular Expressions to Use with the replace Command

Table 14: Common Regular Expressions to Use with the replace Command

Operator	Function
|	Indicates that a match can be one of the two terms on either side of the pipe.
^	Used at the beginning of an expression, denotes where a match should begin.
$	Used at the end of an expression, denotes that a term must be matched exactly up to the point of the **$** character.
[ ]	Specifies a range of letters or digits to match. To separate the start and end of a range, use a hyphen (-).
( )	Specifies a group of terms to match. Stored as numbered variables. Use for back references as **\1 \2 .... \9**.
*****	0 or more terms.
+	One or more terms.
.	Any character except for a space (" ").
****	A backslash escapes special characters to suppress their special meaning. For example, \. matches **.** (period symbol).
n	Back reference. Matches the numbered variable. This is only used with wildcard (.*), where the wildcard (.*) performs the matching, and the \n extracts the content from the variable and inserts it in the matched pattern.

Copyright © 2014, Juniper Networks, Inc.

Table 14: Common Regular Expressions to Use with the replace Command *(continued)*

Operator	Function
&	Back reference. Matches the entire match.

Table 15 on page 189 lists some replacement examples.

Table 15: Replacement Examples

Command	Result
replace pattern myrouter with router1	Match: **myrouter** Result: **router1**
replace pattern " 192\.168\.(.*)/24" with " 10.2.\1/28"	Match: **192.168.3.4/24** Result: **10.2.3.4/28**
replace pattern " 1.\1" with " abc&def"	Match: **1.1** Result: **abc1.1def**
replace pattern 1.1 with " abc\&def"	Match: **1#1** Result: **abc&def**

Related Documentation

- Using Global Replace in a Junos OS Configuration on page 189
- Example: Using Global Replace in a Junos OS Configuration—Using the \n Back Reference on page 190

Using Global Replace in a Junos OS Configuration

You can make global changes to variables and identifiers in a Junos configuration by using the **replace** configuration mode command. This command replaces a pattern in a configuration with another pattern. For example, you can use this command to find and replace all occurrences of an interface name when a PIC is moved to another slot in the router.

user@host# replacepattern *pattern1* with *pattern2* <upto *n*>

pattern *pattern1* is a text string or regular expression that defines the identifiers and values you want to replace in the configuration.

pattern2 is a text string or regular expression that replaces the identifiers and values located with ***pattern1***.

Juniper Networks uses standard UNIX-style regular expression syntax (as defined in POSIX 1003.2). If the regular expression contains spaces, operators, or wildcard characters,

Copyright © 2014, Juniper Networks, Inc.

enclose the expression in quotation marks. Greedy qualifiers (match as much as possible) are supported. Lazy qualifiers (match as little as possible) are not.

The **upto *n*** option specifies the number of objects replaced. The value of ***n*** controls the total number of objects that are replaced in the configuration (not the total number of times the pattern occurs). Objects at the same hierarchy level (siblings) are replaced first. Multiple occurrences of a pattern within a given object are considered a single replacement. For example, if a configuration contains a **010101** text string, the command **replace pattern 01 with pattern 02 upto 2** replaces **010101** with **020202** (instead of **020201**). Replacement of **010101** with **020202** is considered a single replacement (***n*** = **1**), not three separate replacements (***n*** =**3**).

If you do not specify an **upto** option, all identifiers and values in the configuration that match ***pattern1*** are replaced.

The **replace** command is available in configuration mode at any hierarchy level. All matches are case-sensitive.

Related Documentation

- Common Regular Expressions to Use with the replace Command on page 188
- Example: Using Global Replace in a Junos OS Configuration—Using the \n Back Reference on page 190
- Example: Using Global Replace in a Junos OS Configuration—Replacing an Interface Name on page 191
- Example: Using Global Replace in a Junos OS Configuration—Using the upto Option on page 192
- Using Wildcard Characters in Interface Names on page 187
- Using Keyboard Sequences to Move Around and Edit the Junos OS CLI on page 185

Example: Using Global Replace in a Junos OS Configuration—Using the \n Back Reference

The following example shows how you can use the ***n*** back reference to replace a pattern:

```
[edit]
user@host# show interfaces
xe-0/0/0 {
   unit 0;
}
fe-3/0/1 {
   vlan-tagging;
   unit 0 {
      description "inet6 configuration. IP: 2000::c0a8::1bf5";
      vlan-id 100;
      family inet {
         address 17.10.1.1/24;
      }
      family inet6 {
         address 2000::c0a8:1bf5/3;
      }
```


Copyright © 2014, Juniper Networks, Inc.

```
    }
  }
  [edit]
  user@host# replace pattern "(.*):1bf5" with "\11bf5"
  [edit]
  user@host# show interfaces
  xe-0/0/0 {
    unit 0;
  }
  fe-3/0/1 {
    vlan-tagging;
    unit 0 {
      description "inet6 configuration. IP: 2000::c0a8:1bf5";
      vlan-id 100;
      family inet {
        address 17.10.1.1/24;
      }
      family inet6 {
        address 2000::c0a8:1bf5/3;
      }
    }
  }
```

The pattern **2000::c0a8::1bf5** is replaced with **2000::c0a8:1bf5**.

Related Documentation

- Example: Using Global Replace in a Junos OS Configuration—Replacing an Interface Name on page 191
- Using Global Replace in a Junos OS Configuration on page 189

Example: Using Global Replace in a Junos OS Configuration—Replacing an Interface Name

The following example shows how you can replace an interface name in a configuration:

```
  [edit]
  user@host# show
  protocols {
    ospf {
      area 0.0.0.0 {
        interface so-0/0/0 {
          hello-interval 5;
        }
      }
    }
  }
  [edit]
  user@host# replace so-0/0/0 with so-1/1/0
  [edit]
  user@host# show
  protocols {
    ospf {
      area 0.0.0.0 {
        interface so-1/1/0 {
          hello-interval 5;
```

Copyright © 2014, Juniper Networks, Inc.

```
        }
      }
    }
  }
```

Related Documentation

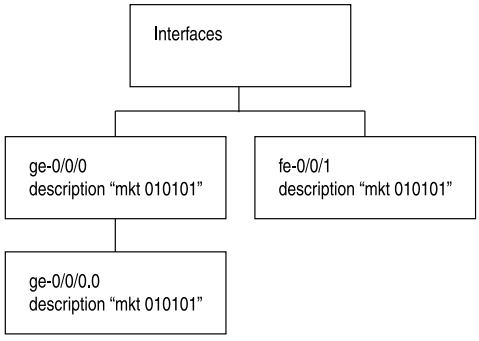
- Example: Using Global Replace in a Junos OS Configuration—Using the upto Option on page 192
- Using Global Replace in a Junos OS Configuration on page 189

Example: Using Global Replace in a Junos OS Configuration—Using the upto Option

Consider the hierarchy shown in Figure 14 on page 192. The text string **010101** appears in three places: the description sections of **ge-0/0/0**, **ge-0/0/0.0**, and **fe-0/0/1**. These three instances are three objects. The following example shows how you can use the **upto** option to perform replacements in a JUNOS configuration:

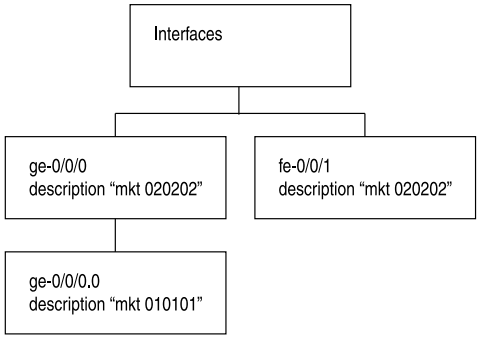
Figure 14: Replacement by Object

Current Configuration:



user@host > **replace pattern 01 with pattern 02 upto 2**

Resulting Configuration:



g017228

An **upto 2** option in the **replace** command converts **01** to **02** for two object instances. The objects under the main interfaces **ge-0/0/0** and **fe-0/0/1** will be replaced first (since these are siblings in the hierarchy level). Because of the **upto 2** restriction, the **replace** command replaces patterns in the first and second instance in the hierarchy (siblings), but not the third instance (child of the first instance).

Copyright © 2014, Juniper Networks, Inc.

```
user@host# show interfaces
ge-0/0/0 {
   description "mkt 010101"; #First instance in the hierarchy
   unit 0 {
      description "mkt 010101"; #Third instance in the hierarchy (child of the first
      instance)
   }
}
fe-0/0/1 {
   description "mkt 010101"; #second instance in the hierarchy (sibling of the first
   instance)
   unit 0 {
      family inet {
         address 200.200.20.2/24;
      }
   }
}
[edit]
user@host# replace pattern 01 with 02 upto 2
[edit]
user@host# commit
commit complete

[edit]
user@host# show interfaces
ge-0/0/0 {
   description "mkt 020202"; #First instance in the hierarchy
   unit 0 {
      description "mkt 010101"; #Third instance in the hierarchy (child of the first
      instance)
   }
}
fe-0/0/1 {
   description "mkt 020202"; #second instance in the hierarchy (sibling of the first
   instance)
   unit 0 {
      family inet {
         address 200.200.20.2/24;
      }
   }
}
```

Related Documentation

- Using Global Replace in a Junos OS Configuration on page 189

Using Regular Expressions to Delete Related Items from a Junos OS cConfiguration

The Junos OS command-line interface (CLI) enables you to delete related configuration items simultaneously, such as channelized interfaces or static routes, by using a single command and regular expressions. Deleting a statement or an identifier effectively “unconfigures” the functionality associated with that statement or identifier, returning that functionality to its default condition.

Copyright © 2014, Juniper Networks, Inc.

You can only delete certain parts of the configuration where you normally put multiple items, for example, interfaces. However, you cannot delete "groups" of different items; for example:

```
user@host# show system services
ftp;
rlogin;
rsh;
ssh {
    root-login allow;
}
telnet;
[edit]
user@host# wildcard delete system services *
syntax error.
```

When you delete a statement, the statement and all its subordinate statements and identifiers are removed from the configuration.

To delete related configuration items, issue the **wildcard** configuration mode command with the **delete** option and specify the statement path, the items to be summarized with a regular expression, and the regular expression.

```
user@host# wildcard delete <statement-path> <identifier> <regular-expression>
```

NOTE: When you use the wildcard command to delete related configuration items, the regular expression must be the final statement.

If the Junos OS matches more than eight related items, the CLI displays only the first eight items.

Deleting Interfaces from the Configuration

Delete multiple T1 interfaces in the range from **t1-0/0/0:0** through **t1-0/0/0:23**:

```
user@host# wildcard delete interfaces t1-0/0/0:.*
matched: t1-0/0/0:0
matched: t1-0/0/0:1
matched: t1-0/0/0:2
Delete 3 objects? [yes,no] (no) no
```


Copyright © 2014, Juniper Networks, Inc.

Deleting Routes from the Configuration

Delete static routes in the range from **172.0.0.0** to **172.255.0.0**:

```
user@host# wildcard delete routing-options static route 172.*
matched: 172.16.0.0/12
matched: 172.16.14.0/24
matched: 172.16.100.0/24
matched: 172.16.128.0/19
matched: 172.16.160.0/24
matched: 172.17.12.0/23
matched: 172.17.24.0/23
matched: 172.17.28.0/23
...
Delete 13 objects? [yes,no] (no)
```

Related Documentation

- Disabling Inheritance of a Junos OS Configuration Group on page 204

Copyright © 2014, Juniper Networks, Inc.

Copyright © 2014, Juniper Networks, Inc.

CHAPTER 10

Using Configuration Groups to Quickly Configure Devices

- Understanding Junos OS Configuration Groups on page 198
- Creating a Junos OS Configuration Group on page 199
- Applying a Junos OS Configuration Group on page 201
- Example: Configuring and Applying Junos OS Configuration Groups on page 202
- Example: Creating and Applying Configuration Groups on a TX Matrix Router on page 203
- Disabling Inheritance of a Junos OS Configuration Group on page 204
- Using Wildcards with Configuration Groups on page 206
- Example: Configuring Sets of Statements with Configuration Groups on page 209
- Example: Configuring Interfaces Using Junos OS Configuration Groups on page 210
- Example: Configuring a Consistent IP Address for the Management Interface on page 212
- Example: Configuring Peer Entities on page 214
- Establishing Regional Configurations on page 215
- Selecting Wildcard Names on page 217
- Example: Referencing the Preset Statement From the Junos OS defaults Group on page 218
- Example: Viewing Default Statements That Have Been Applied to the Configuration on page 219
- Using Conditions to Apply Configuration Groups Overview on page 219
- Example: Configuring Conditions for Applying Configuration Groups on page 220
- Improving Commit Time When Using Configuration Groups on page 222
- Example: Improving Commit Time When Using Configuration Groups on page 223
- Using Junos OS Defaults Groups on page 224
- Set Up Routing Engine Configuration Groups on page 225

Copyright © 2014, Juniper Networks, Inc.

Understanding Junos OS Configuration Groups

This topic provides an overview of the configuration groups feature and the inheritance model in Junos OS, and contains the following sections:

- Configuration Groups Overview on page 198
- Inheritance Model on page 198
- Configuring Configuration Groups on page 198

Configuration Groups Overview

The configuration groups feature in Junos OS enables you to create a group containing configuration statements and to direct the inheritance of that group's statements in the rest of the configuration. The same group can be applied to different sections of the configuration, and different sections of one group's configuration statements can be inherited in different places in the configuration.

Configuration groups enable you to create smaller, more logically constructed configuration files, making it easier to configure and maintain Junos OS. For example, you can group statements that are repeated in many places in the configuration, such as when configuring interfaces, and thereby limit updates to just the group.

You can also use wildcards in a configuration group to allow configuration data to be inherited by any object that matches a wildcard expression.

The configuration group mechanism is separate from the grouping mechanisms used elsewhere in the configuration, such as BGP groups. Configuration groups provide a generic mechanism that can be used throughout the configuration but that are known only to the Junos OS CLI. The individual software processes that perform the actions directed by the configuration receive the expanded form of the configuration—they have no knowledge of configuration groups.

Inheritance Model

Configuration groups use true inheritance, which involves a dynamic, ongoing relationship between the source of the configuration data and the target of that data. Data values changed in the configuration group are automatically inherited by the target. The target does not need to contain the inherited information, although the inherited values can be overridden in the target without affecting the source from which they were inherited.

This inheritance model allows you to see only the instance-specific information without seeing the inherited details. A command pipe in configuration mode allows you to display the inherited data.

Configuring Configuration Groups

For areas of your configuration to inherit configuration statements, you must first put the statements into a configuration group and then apply that group to the levels in the configuration hierarchy that require the statements.

Copyright © 2014, Juniper Networks, Inc.

To configure configuration groups and inheritance, you can include the **groups** statement at the **[edit]** hierarchy level:

```
[edit]
groups {
  group-name {
    configuration-data;
  }
}
```

Include the **apply-groups [** *group-names* **]** statement anywhere in the configuration where the configuration statements contained in a configuration group are needed.

Related Documentation

- Creating a Junos OS Configuration Group on page 199

Creating a Junos OS Configuration Group

To create a configuration group, include the **groups** statement at the **[edit]** hierarchy level:

```
[edit]
groups {
  group-name {
    configuration-data;
  }
  lccn-re0 {
    configuration-data;
  }
  lccn-re1 {
    configuration-data;
  }
}
```

group-name is the name of a configuration group. You can configure more than one configuration group by specifying multiple ***group-name*** statements. However, you cannot use the prefix **junos-** in a group name because it is reserved for use by Junos OS. Similarly, the configuration group **juniper-ais** is reserved exclusively for Juniper Advanced Insight Solutions (AIS)-related configuration. For more information on the **juniper-ais** configuration group, see the Juniper Networks *Advanced Insight Solutions Guide*.

One reason for the naming restriction is a configuration group called **junos-defaults**. This preset configuration group is applied to the configuration automatically. You cannot modify or remove the **junos-defaults** configuration group. For more information about the Junos default configuration group, see "Using Junos OS Defaults Groups" on page 224.

On routers that support multiple Routing Engines, you can also specify two special group names:

- **re0**—Configuration statements applied to the Routing Engine in slot 0.
- **re1**—Configuration statements applied to the Routing Engine in slot 1.

Copyright © 2014, Juniper Networks, Inc.

The configuration specified in group **re0** is only applied if the current Routing Engine is in slot 0; likewise, the configuration specified in group **re1** is only applied if the current Routing Engine is in slot 1. Therefore, both Routing Engines can use the same configuration file, each using only the configuration statements that apply to it. Each **re0** or **re1** group contains at a minimum the configuration for the hostname and the management interface (fxp0). If each Routing Engine uses a different management interface, the group also should contain the configuration for the backup router and static routes.

In addition, the TX Matrix router supports group names for the Routing Engines in each T640 router attached to the routing matrix. Providing special group names for all Routing Engines in the routing matrix allows you to configure the individual Routing Engines in each T640 router differently. Parameters that are not configured at the **[edit groups]** hierarchy level apply to all Routing Engines in the routing matrix.

configuration-data contains the configuration statements applied elsewhere in the configuration with the **apply-groups** statement. To have a configuration inherit the statements in a configuration group, include the **apply-groups** statement. For information about the **apply-groups** statement, see "Applying a Junos OS Configuration Group" on page 201.

The group names for Routing Engines on the TX Matrix router have the following formats:

- **lcc*n*-re0**—Configuration statements applied to the Routing Engine in slot 0 in a specified T640 router.
- **lcc*n*-re1**—Configuration statements applied to the Routing Engine in slot 1 in a specified T640 router.

n identifies the T640 router and can be from 0 through 3. For example, to configure Routing Engine 1 properties for **lcc3**, you include statements at the **[edit groups lcc3–re1]** hierarchy level. For information about the TX Matrix router and routing matrix, see the *Junos OS Administration Library for Routing Devices*.

NOTE: **The management Ethernet interface used for the TX Matrix Plus router, T1600 or T4000 routers in a routing matrix, and PTX Series Packet Transport Routers, is em0. Junos OS automatically creates the router's management Ethernet interface, em0.**

Related Documentation

- Applying a Junos OS Configuration Group on page 201
- Using Junos OS Defaults Groups on page 224
- Understanding Junos OS Configuration Groups on page 198
- Disabling Inheritance of a Junos OS Configuration Group on page 204
- Using Wildcards with Configuration Groups on page 206
- Example: Configuring Sets of Statements with Configuration Groups on page 209

Copyright © 2014, Juniper Networks, Inc.

Applying a Junos OS Configuration Group

To have a Junos configuration inherit the statements from a configuration group, include the **apply-groups** statement:

apply-groups [*group-names*];

If you specify more than one group name, list them in order of inheritance priority. The configuration data in the first group takes priority over the data in subsequent groups.

For routers that support multiple Routing Engines, you can specify **re0** and **re1** group names. The configuration specified in group **re0** is only applied if the current Routing Engine is in slot 0; likewise, the configuration specified in group **re1** is only applied if the current Routing Engine is in slot 1. Therefore, both Routing Engines can use the same configuration file, each using only the configuration statements that apply to it. Each **re0** or **re1** group contains at a minimum the configuration for the hostname and the management interface (**fxp0**). If each Routing Engine uses a different management interface, the group also should contain the configuration for the backup router and static routes.

NOTE: The management Ethernet interface used for the TX Matrix Plus router, T1600 or T4000 routers in a routing matrix, and PTX Series Packet Transport Routers, is em0.

You can include only one **apply-groups** statement at each specific level of the configuration hierarchy. The **apply-groups** statement at a specific hierarchy level lists the configuration groups to be added to the containing statement's list of configuration groups.

Values specified at the specific hierarchy level override values inherited from the configuration group.

Groups listed in nested **apply-groups** statements take priority over groups in outer statements. In the following example, the BGP neighbor **10.0.0.1** inherits configuration data from group **one** first, then from groups **two** and **three**. Configuration data in group **one** overrides data in any other group. Data from group **ten** is used only if a statement is not contained in any other group.

```
apply-groups [ eight nine ten ];
protocols {
    apply-groups seven;
    bgp {
        apply-groups [ five six ];
        group some-bgp-group {
            apply-groups four;
            neighbor 10.0.0.1 {
                apply-groups [ one two three ];
            }
        }
    }
}
```

Copyright © 2014, Juniper Networks, Inc.

When you configure a group defined for the root level—that is, in the default logical system–you cannot successfully apply that group to a nondefault logical system under the **[edit logical-systems *logical-system-name*]** hierarchy level. Although the router accepts the commit if you apply the group, the configuration group does not take effect for the nondefault logical system. You can instead create an additional configuration group at the root level and apply it within the logical system. Alternatively, you can modify the original group so that it includes configuration for both the default and nondefault logical system hierarchy levels.

Related Documentation

- Example: Configuring and Applying Junos OS Configuration Groups on page 202
- Disabling Inheritance of a Junos OS Configuration Group on page 204
- Creating a Junos OS Configuration Group on page 199
- Using Wildcards with Configuration Groups on page 206
- Example: Configuring Sets of Statements with Configuration Groups on page 209

Example: Configuring and Applying Junos OS Configuration Groups

In this example, the SNMP configuration is divided between the group **basic** and the normal configuration hierarchy.

There are a number of advantages to placing the system-specific configuration (SNMP contact) into a configuration group and thus separating it from the normal configuration hierarchy—the user can replace (using the **load replace** command) either section without discarding data from the other.

In addition, setting a contact for a specific box is now possible because the group data would be hidden by the router-specific data.

```
[edit]
groups {
   basic { # User-defined group name
      snmp { # This group contains some SNMP data
         contact "My Engineering Group";
         community BasicAccess {
            authorization read-only;
         }
      }
   }
}
apply-groups basic; # Enable inheritance from group "basic"
snmp { # Some normal (non-group) configuration
   location "West of Nowhere";
}
```

This configuration is equivalent to the following:

```
[edit]
snmp {
   location "West of Nowhere";
   contact "My Engineering Group";
```


Copyright © 2014, Juniper Networks, Inc.

```
    community BasicAccess {
      authorization read-only;
    }
  }
```

For information about how to disable inheritance of a configuration group, see "Disabling Inheritance of a Junos OS Configuration Group" on page 204.

Related Documentation

- Example: Creating and Applying Configuration Groups on a TX Matrix Router on page 203
- Example: Configuring Interfaces Using Junos OS Configuration Groups on page 210
- Example: Configuring Peer Entities on page 214
- Example: Referencing the Preset Statement From the Junos OS defaults Group on page 218
- Example: Viewing Default Statements That Have Been Applied to the Configuration on page 219
- Example: Configuring Sets of Statements with Configuration Groups on page 209
- Example: Configuring a Consistent IP Address for the Management Interface on page 212
- Creating a Junos OS Configuration Group on page 199

Example: Creating and Applying Configuration Groups on a TX Matrix Router

The following example shows how to configure and apply configuration groups on a TX Matrix Router:

```
[edit]
groups {
  re0 { # Routing Engine 0 on TX Matrix router
    system {
      host-name hostname;
      backup-router ip-address;
    }
    interfaces {
      fxp0 {
        unit 0 {
          family inet {
            address ip-address;
          }
        }
      }
    }
  }
  re1 { # Routing Engine 1 on TX Matrix router
    system {
      host-name hostname;
      backup-router ip-address;
    }
    interfaces {
      fxp0 {
        unit 0 {
```

Copyright © 2014, Juniper Networks, Inc.

```
            family inet {
              address ip-address;
            }
          }
        }
      }
    }
    lcc0-re0 { # Routing Engine 0 on T640 router numbered 0
      system {
        host-name hostname;
        backup-router ip-address;
      }
      interfaces {
        fxp0 {
          unit 0 {
            family inet {
              address ip-address;
            }
          }
        }
      }
    }
    lcc0-re1 { # Routing Engine 1 on T640 router numbered 0
      system {
        host-name hostname;
        backup-router ip-address;
      }
      interfaces {
        fxp0 {
          unit 0 {
            family inet {
              address ip-address;
            }
          }
        }
      }
    }
  }
  apply-groups [ re0 re1 lcc0-re0 lcc0-re1 ];
```

Related Documentation

- Example: Configuring and Applying Junos OS Configuration Groups on page 202
- Creating a Junos OS Configuration Group on page 199

Disabling Inheritance of a Junos OS Configuration Group

To disable inheritance of a configuration group at any level except the top level of the hierarchy, include the **apply-groups-except** statement:

```
apply-groups-except [ group-names ];
```

This statement is useful when you use the **apply-group** statement at a specific hierarchy level but also want to override the values inherited from the configuration group for a specific parameter.

Copyright © 2014, Juniper Networks, Inc.

Example: Disabling Inheritance on Interface so-1/1/0

In the following example, the **apply-groups** statement is applied globally at the interfaces level. The **apply-groups-except** statement is also applied at interface **so-1/1/0** so that it uses the default values for the **hold-time** and **link-mode** statements.

```
[edit]
groups { # "groups" is a top-level statement
  global { # User-defined group name
    interfaces {
      <*> {
        hold-time down 640;
        link-mode full-duplex;
      }
    }
  }
}
apply-groups global;
interfaces {
  so-1/1/0 {
    apply-groups-except global;  # Disables inheritance from group "global"
    # so-1/1/0 uses default value for “hold-time”
    # and "link-mode"
  }
}
```

For information about applying a configuration group, see “Applying a Junos OS Configuration Group” on page 201.

Configuration groups can add some confusion regarding the actual values used by the router, because configuration data can be inherited from configuration groups. To view the actual values used by the router, use the **display inheritance** command after the pipe (|) in a **show** command. This command displays the inherited statements at the level at which they are inherited and the group from which they have been inherited.

```
[edit]
user@host# show | display inheritance
snmp {
  location "West of Nowhere";
  ##
  ## 'My Engineering Group' was inherited from group 'basic'
  ##
  contact "My Engineering Group";
  ##
  ## 'BasicAccess' was inherited from group 'basic'
  ##
  community BasicAccess {
    ##
    ## 'read-only' was inherited from group 'basic'
    ##
    authorization read-only;
  }
}
```

To display the expanded configuration (the configuration, including the inherited statements) without the ## lines, use the **except** command after the pipe in a **show** command:

Copyright © 2014, Juniper Networks, Inc.

```
[edit]
user@host# show | display inheritance | except ##
snmp {
   location "West of Nowhere";
   contact "My Engineering Group";
   community BasicAccess {
      authorization read-only;
   }
}
```

NOTE: Using the **display inheritance | except ##** option removes all the lines with **##**. Therefore, you might also not be able to view information about passwords and other important data where **##** is used. To view the complete configuration details with all the information without just the comments marked with **##**, use the **no-comments** option with the **display inheritance** command:

```
[edit]
user@host# show | display inheritance no-comments
snmp {
   location "West of Nowhere";
   contact "My Engineering Group";
   community BasicAccess {
      authorization read-only;
   }
}
```

Related Documentation

- Applying a Junos OS Configuration Group on page 201
- Understanding Junos OS Configuration Groups on page 198

Using Wildcards with Configuration Groups

You can use wildcards to identify names and allow one statement to provide data for a variety of statements. For example, grouping the configuration of the **sonet-options** statement over all SONET/SDH interfaces or the dead interval for OSPF over all Asynchronous Transfer Mode (ATM) interfaces simplifies configuration files and eases their maintenance.

Using wildcards in normal configuration data is done in a style that is consistent with that used with traditional UNIX shell wildcards. In this style, you can use the following metacharacters:

- Asterisk (*****)—Matches any string of characters.
- Question mark (**?**)—Matches any single character.
- Open bracket (**[**)—Introduces a character class.
- Close bracket (**]**)—Indicates the end of a character class. If the close bracket is missing, the open bracket matches a **[** rather than introduce a character class.

Copyright © 2014, Juniper Networks, Inc.

- A character class matches any of the characters between the square brackets. Within a configuration group, an interface name that includes a character class must be enclosed in quotation marks.
- Hyphen (**-**)—Specifies a range of characters.
- Exclamation point (**!**)—The character class can be complemented by making an exclamation point the first character of the character class. To include a close bracket (**]**) in a character class, make it the first character listed (after the **!**, if any). To include a minus sign, make it the first or last character listed.

Wildcarding in configuration groups follows the same rules, but any term using a wildcard pattern must be enclosed in angle brackets <*pattern*> to differentiate it from other wildcarding in the configuration file.

```
[edit]
groups {
   sonet-default {
      interfaces {
         <so-*> {
            sonet-options {
               payload-scrambler;
               rfc-2615;
            }
         }
      }
   }
}
```

Wildcard expressions match (and provide configuration data for) existing statements in the configuration that match their expression only. In the previous example, the expression **<so-*>** passes its **sonet-options** statement to any interface that matches the expression **so-***.

The following example shows how to specify a range of interfaces:

```
[edit]
groups {
   gigabit-ethernet-interfaces {
      interfaces {
         "<ge-1/2/[5-8]>" {
            description "These interfaces reserved for Customer ABC";
         }
      }
   }
}
```

Angle brackets allow you to pass normal wildcarding through without modification. In any matching within the configuration, whether it is done with or without wildcards, the first item encountered in the configuration that matches is used. In the following example, data from the wildcarded BGP groups is inherited in the order in which the groups are listed. The preference value from **<*a*>** overrides the preference in **<*b*>**, just as the **p** value from **<*c*>** overrides the one from **<*d*>**. Data values from any of these groups override the data values from **abcd**.

Copyright © 2014, Juniper Networks, Inc.

```
[edit]
user@host# show
groups {
  one {
    protocols {
      bgp {
        group <*a*> {
          preference 1;
        }
        group <*b*> {
          preference 2;
        }
        group <*c*> {
          out-delay 3;
        }
        group <*d*> {
          out-delay 4;
        }
        group abcd {
          preference 10;
          hold-time 10;
          out-delay 10;
        }
      }
    }
  }
}
protocols {
  bgp {
    group abcd {
      apply-groups one;
    }
  }
}
[edit]
user@host# show | display inheritance
protocols {
  bgp {
    group abcd {
      ##
      ## '1' was inherited from group 'one'
      ##
      preference 1;
      ##
      ## '10' was inherited from group 'one'
      ##
      hold-time 10;
      ##
      ## '3' was inherited from group 'one'
      ##
      out-delay 3;
    }
  }
}
```


Copyright © 2014, Juniper Networks, Inc.

Related Documentation

- Selecting Wildcard Names on page 217
- Applying a Junos OS Configuration Group on page 201
- Creating a Junos OS Configuration Group on page 199
- Understanding Junos OS Configuration Groups on page 198

Example: Configuring Sets of Statements with Configuration Groups

When sets of statements exist in configuration groups, all values are inherited. For example:

```
[edit]
user@host# show
groups {
    basic {
        snmp {
            interface so-1/1/1.0;
        }
    }
}
apply-groups basic;
snmp {
    interface so-0/0/0.0;
}
[edit]
user@host# show | display inheritance
snmp {
    ##
    ## 'so-1/1/1.0' was inherited from group 'basic'
    ##
    interface [ so-0/0/0.0 so-1/1/1.0 ];
}
```

For sets that are not displayed within brackets, all values are also inherited. For example:

```
[edit]
user@host# show
groups {
    worldwide {
        system {
            name-server {
                10.0.0.100;
                10.0.0.200;
            }
        }
    }
}
apply-groups worldwide;
system {
    name-server {
        10.0.0.1;
        10.0.0.2;
    }
```

Copyright © 2014, Juniper Networks, Inc.

```
}
[edit]
user@host# show | display inheritance
system {
  name-server {
    ##
    ## '10.0.0.100' was inherited from group 'worldwide'
    ##
    10.0.0.100;
    ##
    ## '10.0.0.200' was inherited from group 'worldwide'
    ##
    10.0.0.200;
    10.0.0.1;
    10.0.0.2;
  }
}
```

Related Documentation

- Understanding Junos OS Configuration Groups on page 198
- Creating a Junos OS Configuration Group on page 199
- Applying a Junos OS Configuration Group on page 201

Example: Configuring Interfaces Using Junos OS Configuration Groups

You can use configuration groups to separate the common interface media parameters from the interface-specific addressing information. The following example places configuration data for ATM interfaces into a group called **atm-options**:

```
[edit]
user@host# show
groups {
  atm-options {
    interfaces {
      <at-*> {
        atm-options {
          vpi 0 maximum-vcs 1024;
        }
        unit <*> {
          encapsulation atm-snap;
          point-to-point;
          family iso;
        }
      }
    }
  }
}
apply-groups atm-options;
interfaces {
  at-0/0/0 {
    unit 100 {
      vci 0.100;
      family inet {
        address 10.0.0.100/30;
```


Copyright © 2014, Juniper Networks, Inc.

```
        }
      }
      unit 200 {
        vci 0.200;
        family inet {
          address 10.0.0.200/30;
        }
      }
    }
}
[edit]
user@host# show | display inheritance
interfaces {
  at-0/0/0 {
    ##
    ## "atm-options" was inherited from group "atm-options"
    ##
    atm-options {
      ##
      ## "1024" was inherited from group "atm-options"
      ##
      vpi 0 maximum-vcs 1024;
    }
    unit 100 {
      ##
      ## "atm-snap" was inherited from group "atm-options"
      ##
      encapsulation atm-snap;
      ##
      ## "point-to-point" was inherited from group "atm-options"
      ##
      point-to-point;
      vci 0.100;
      family inet {
        address 10.0.0.100/30;
      }
      ##
      ## "iso" was inherited from group "atm-options"
      ##
      family iso;
    }
    unit 200 {
      ##
      ## "atm-snap" was inherited from group "atm-options"
      ##
      encapsulation atm-snap;
      ##
      ## "point-to-point" was inherited from group "atm-options"
      ##
      point-to-point;
      vci 0.200;
      family inet {
        address 10.0.0.200/30;
      }
      ##
      ## "iso" was inherited from group "atm-options"
```

```
        ##
        family iso;
      }
    }
  }
[edit]
user@host# show | display inheritance | except ##
interfaces {
  at-0/0/0 {
    atm-options {
      vpi 0 maximum-vcs 1024;
    }
    unit 100 {
      encapsulation atm-snap;
      point-to-point;
      vci 0.100;
      family inet {
        address 10.0.0.100/30;
      }
      family iso;
    }
    unit 200 {
      encapsulation atm-snap;
      point-to-point;
      vci 0.200;
      family inet {
        address 10.0.0.200/30;
      }
      family iso;
    }
  }
}
```

Related Documentation

- Understanding Junos OS Configuration Groups on page 198
- Creating a Junos OS Configuration Group on page 199
- Interface Naming Conventions Used in the Junos OS Operational Commands on page 158
- Example: Configuring a Consistent IP Address for the Management Interface on page 212

Example: Configuring a Consistent IP Address for the Management Interface

On routers with multiple Routing Engines, each Routing Engine is configured with a separate IP address for the management interface (**fxp0**). To access the master Routing Engine, you must know which Routing Engine is active and use the appropriate IP address.

Optionally, for consistent access to the master Routing Engine, you can configure an additional IP address and use this address for the management interface regardless of which Routing Engine is active. This additional IP address is active only on the management interface for the master Routing Engine. During switchover, the address moves to the new master Routing Engine.

Copyright © 2014, Juniper Networks, Inc.

In the following example, address **10.17.40.131** is configured for both Routing Engines and includes a **master-only** statement. With this configuration, the **10.17.40.131** address is active only on the master Routing Engine. The address remains consistent regardless of which Routing Engine is active. Address **10.17.40.132** is assigned to **fxp0** on **re0**, and **10.17.40.133** is assigned to **fxp0** on **re1**.

```
[edit groups re0 interfaces fxp0]
unit 0 {
   family inet {
      address 10.17.40.131/25 {
         master-only;
      }
      address 10.17.40.132/25;
   }
}
[edit groups re1 interfaces fxp0]
unit 0 {
   family inet {
      address 10.17.40.131/25 {
         master-only;
      }
      address 10.17.40.133/25;
   }
}
```

This feature is available on all routers that include dual Routing Engines. On a routing matrix composed of the TX Matrix router, this feature is applicable to the switch-card chassis (SCC) only. Likewise, on a routing matrix composed of a TX Matrix Plus router, this feature is applicable to the switch-fabric chassis (SFC) only.

NOTE:

- If you configure the same IP address for a management interface or internal interface such as **fxp0** and an external physical interface such as **ge-0/0/1**, when graceful Routing Engine switchover (GRES) is enabled, the CLI displays an appropriate commit error message that identical addresses have been found on the private and public interfaces. In such cases, you must assign unique IP addresses for the two interfaces that have duplicate addresses.
- The management Ethernet interface used for the TX Matrix Plus router, T1600 routers in a routing matrix, and PTX Series Packet Transport Routers, is **em0**. Junos OS automatically creates the router's management Ethernet interface, **em0**.

Related Documentation

- Understanding Junos OS Configuration Groups on page 198
- Creating a Junos OS Configuration Group on page 199
- Example: Configuring Interfaces Using Junos OS Configuration Groups on page 210

Copyright © 2014, Juniper Networks, Inc.

Example: Configuring Peer Entities

In this example, we create a group **some-isp** that contains configuration data relating to another Internet service provider (ISP). We can then insert **apply-group** statements at any point to allow any location in the configuration hierarchy to inherit this data.

```
[edit]
user@host# show
groups {
    some-isp {
        interfaces {
            <xe-*> {
                gigether-options {
                    flow-control;
                }
            }
        }
        protocols {
            bgp {
                group <*> {
                    neighbor <*> {
                        remove-private;
                    }
                }
            }
            pim {
                interface <*> {
                    version 1;
                }
            }
        }
    }
}
interfaces {
    xe-0/0/0 {
        apply-groups some-isp;
        unit 0 {
            family inet {
                address 10.0.0.1/24;
            }
        }
    }
}
protocols {
    bgp {
        group main {
            neighbor 10.254.0.1 {
                apply-groups some-isp;
            }
        }
    }
    pim {
        interface xe-0/0/0.0 {
            apply-groups some-isp;
```


Copyright © 2014, Juniper Networks, Inc.

```
        }
      }
    }
    [edit]
    user@host# show | display inheritance
    interfaces {
      xe-0/0/0 {
        ##
        ## "gigether-options" was inherited from group "some-isp"
        ##
        gigether-options {
          ##
          ## "flow-control" was inherited from group "some-isp"
          ##
          flow-control;
        }
        unit 0 {
          family inet {
            address 10.0.0.1/24;
          }
        }
      }
    }
    protocols {
      bgp {
        group main {
          neighbor 10.254.0.1 {
            ##
            ## "remove-private" was inherited from group "some-isp"
            ##
            remove-private;
          }
        }
      }
      pim {
        interface xe-0/0/0.0 {
          ##
          ## "1" was inherited from group "some-isp"
          ##
          version 1;
        }
      }
    }
```

Related Documentation

- Understanding Junos OS Configuration Groups on page 198
- Creating a Junos OS Configuration Group on page 199
- Establishing Regional Configurations on page 215

Establishing Regional Configurations

In this example, one group is populated with configuration data that is standard throughout the company, while another group contains regional deviations from this standard:

```
[edit]
user@host# show
groups {
  standard {
    interfaces {
      <t3-*> {
        t3-options {
          compatibility-mode larscom subrate 10;
          idle-cycle-flag ones;
        }
      }
    }
  }
  northwest {
    interfaces {
      <t3-*> {
        t3-options {
          long-buildout;
          compatibility-mode kentrox;
        }
      }
    }
  }
}
apply-groups standard;
interfaces {
  t3-0/0/0 {
    apply-groups northwest;
  }
}
[edit]
user@host# show | display inheritance
interfaces {
  t3-0/0/0 {
    ##
    ## "t3-options" was inherited from group "northwest"
    ##
    t3-options {
      ##
      ## "long-buildout" was inherited from group "northwest"
      ##
      long-buildout;
      ##
      ## "kentrox" was inherited from group "northwest"
      ##
      compatibility-mode kentrox;
      ##
      ## "ones" was inherited from group "standard"
      ##
      idle-cycle-flag ones;
    }
  }
}
```


Copyright © 2014, Juniper Networks, Inc.

Related Documentation

- Understanding Junos OS Configuration Groups on page 198
- Example: Configuring Peer Entities on page 214

Selecting Wildcard Names

You can combine wildcarding and thoughtful use of names in statements to tailor statement values:

```
[edit]
user@host# show
groups {
   mpls-conf {
      protocols {
         mpls {
            label-switched-path <*-major> {
               retry-timer 5;
               bandwidth 155m;
               optimize-timer 60;
            }
            label-switched-path <*-minor> {
               retry-timer 15;
               bandwidth 64k;
               optimize-timer 120;
            }
         }
      }
   }
}
apply-groups mpls-conf;
protocols {
   mpls {
      label-switched-path metro-major {
         to 10.0.0.10;
      }
      label-switched-path remote-minor {
         to 10.0.0.20;
      }
   }
}
[edit]
user@host# show | display inheritance
protocols {
   mpls {
      label-switched-path metro-major {
         to 10.0.0.10;
         ##
         ## "5" was inherited from group "mpls-conf"
         ##
         retry-timer 5;
         ## "155m" was inherited from group "mpls-conf"
         ##
         bandwidth 155m;
         ##
```

Copyright © 2014, Juniper Networks, Inc.

```
            ## "60" was inherited from group "mpls-conf"
            ##
            optimize-timer 60;
          }
          label-switched-path remote-minor {
            to 10.0.0.20;
            ##
            ## "15" was inherited from group "mpls-conf"
            ##
            retry-timer 15;
            ##
            ## "64k" was inherited from group "mpls-conf"
            ##
            bandwidth 64k;
            ##
            ## "120" was inherited from group "mpls-conf"
            ##
            optimize-timer 120;
          }
        }
      }
```

Related Documentation

- Using Wildcards with Configuration Groups on page 206

Example: Referencing the Preset Statement From the Junos OS defaults Group

The following example is a preset statement from the Junos defaults group that is available for FTP in a stateful firewall:

```
[edit]
groups {
  junos-defaults {
    applications {
      application junos-ftp {# Use FTP default configuration
      application-protocol ftp;
      protocol tcp;
      destination-port 21;
    }
  }
}
```

To reference a preset Junos default statement from the Junos defaults group, include the **junos-*default-name*** statement at the applicable hierarchy level. For example, to reference the Junos default statement for FTP in a stateful firewall, include the **junos-ftp** statement at the **[edit services stateful-firewall rule my-rule term my-term from applications]** hierarchy level:

```
[edit]
services {
  stateful-firewall {
    rule my-rule {
      term my-term {
        from {
          applications junos-ftp; #Reference predefined statement, junos-ftp
```


Copyright © 2014, Juniper Networks, Inc.

```
        }
      }
    }
  }
}
```

Related Documentation

- Example: Viewing Default Statements That Have Been Applied to the Configuration on page 219
- Using Junos OS Defaults Groups on page 224
- Understanding Junos OS Configuration Groups on page 198
- Creating a Junos OS Configuration Group on page 199

Example: Viewing Default Statements That Have Been Applied to the Configuration

To view the Junos defaults that have been applied to the configuration, issue the **show | display inheritance defaults** command. For example, to view the inherited Junos defaults at the **[edit system ports]** hierarchy level:

```
user@host# show system ports | display inheritance defaults
## ## 'console' was inherited from group 'junos-defaults'
## 'vt100' was inherited from group 'junos-defaults'
## console type vt100;
```

If you choose not to use existing Junos default statements, you can create your own configuration groups manually.

To view the complete configuration information without the comments marked with **##**, use the **no-comments** option with the **display inheritance** command.

Related Documentation

- Creating a Junos OS Configuration Group on page 199
- Configuring Configuration Groups on page 198

Using Conditions to Apply Configuration Groups Overview

You can use the **when** statement at the **[edit groups *group-name*]** hierarchy level to define conditions under which a configuration group should be applied.

You can configure a group to be applied based on the type of chassis, model, or Routing Engine, virtual chassis member, cluster node, and start and optional end time of day or date.

For example, you could use the **when** statement to create a generic configuration group for each type of node and then apply the configuration based on certain node properties, such as chassis or model.

Related Documentation

- Example: Configuring Conditions for Applying Configuration Groups on page 220

Copyright © 2014, Juniper Networks, Inc.

Example: Configuring Conditions for Applying Configuration Groups

This example shows how to configure conditions under which a specified configuration group is to be applied.

- Requirements on page 220
- Overview on page 220
- Configuration on page 220

Requirements

No special configuration beyond device initialization is required before you configure this example.

Overview

You can configure your group configuration data at the **[edit groups *group-name*]** hierarchy level, then use the **when** statement to have the group applied based on conditions including: type of chassis, model, routing-engine, virtual chassis member, cluster node, and start and optional end time of day or date.

If you specify multiple conditions in a single configuration group, all conditions must be met before the configuration group is applied.

You can specify the start time or the time duration for the configuration group to be applied. If only the start time is specified, the configuration group is applied at the specified time and it remains in effect until the time is changed. If the end time is specified, then on each day, the applied configuration group is started and stopped at the specified times.

This example sets conditions in a configuration group, **test1**, such that this group is applied only when all of the following conditions are met: the router is a model MX240 router with chassis type LCC0, with a Routing Engine operating as RE0, is member0 of the virtual chassis on node0, and the configuration group will only be in effect from 9:00 a.m. until 5:00 p.m. each day.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set groups test1 when model mx240
set groups test1 when chassis lcc0
set groups test1 when routing-engine re0
set groups test1 when member member0
set groups test1 when node node0
set groups test1 when time 9 to 5
```


Copyright © 2014, Juniper Networks, Inc.

Step-by-Step Procedure

To configure conditions for configuration group **test1**:

1. Set the condition that identifies the model MX240 router.

   ```
   [edit groups test1 when]
   user@host# set model mx240
   ```

2. Set the condition that identifies the chassis type as LCC0.

   ```
   [edit groups test1 when]
   user@host# set chassis lcc0
   ```

3. Set the condition that identifies the Routing Engine operating as **RE0**.

   ```
   [edit groups test1 when]
   user@host# set routing-engine re0
   ```

4. Set the condition that identifies the virtual chassis **member0**.

   ```
   [edit groups test1 when]
   user@host# set member member0
   ```

5. Set the condition that identifies the cluster **node0**.

   ```
   [edit groups test1 when]
   user@host# set node node0
   ```

6. Set the condition that applies the group only between the hours of 9:00 a.m. and 5:00 p.m. daily.

   ```
   [edit groups test1 when]
   user@host# set time 9 to 5
   ```

   NOTE: The syntax for specifying the time is: time *<start-time>* [to *<end-time>*] using the time format yyyy-mm-dd.hh:mm, hh:mm, or hh.

7. Commit the configuration.

   ```
   user@host# commit
   ```

Results

From configuration mode, confirm your configuration by entering the **show groups test1** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show groups test1
when {
    time 9 to 5;
    chassis lcc0;
    model mx240;
    routing-engine re0;
    member member0;
    node node0;
}
```

Copyright © 2014, Juniper Networks, Inc.

Verification

Confirm that the configuration is working properly.

- Checking Group Inheritance with Conditional Data on page 222

Checking Group Inheritance with Conditional Data

Purpose Verify that conditional data from a configuration group is inherited when applied.

Action The **show | display inheritance** operational command can be issued with the **when** data to display the conditional inheritance. Using this example, you could issue one of these commands to determine that the conditional data was inherited:

```
user@host> show | display inheritance when model mx240
user@host> show | display inheritance when chassis lcc0
user@host> show | display inheritance when routing-engine re0
user@host> show | display inheritance when member member0
user@host> show | display inheritance when node node0
user@host> show | display inheritance when time 9 to 5
```

Related Documentation

- Understanding Junos OS Configuration Groups on page 198
- Creating a Junos OS Configuration Group on page 199
- Applying a Junos OS Configuration Group on page 201
- Using Conditions to Apply Configuration Groups Overview on page 219

Improving Commit Time When Using Configuration Groups

Configuration groups are used for applying configurations across other hierarchies without re-entering configuration data. Some configuration groups specify every configuration detail. Other configuration groups make use of wildcards to configure ranges of data, without detailing each configuration line. Some configurations have an inheritance path that includes a long string of configurations to be applied.

When a configuration that uses configuration groups is committed, the commit process expands and reads all of the configuration data of the group into memory in order to apply the configurations as intended. The commit performance can be negatively impacted if many configuration groups are being applied, especially if the configuration groups use wildcards extensively.

If your system uses many configuration groups that use wildcards, you can configure the **persist-groups-inheritance** statement at the **[edit system commit]** hierarchy level to improve commit time performance.

Using this option allows the system to build the inheritance path for each configuration group inside the database, instead of in the process memory. This can improve commit time performance. However, it can also can increase the database size by up to 22 percent.

Copyright © 2014, Juniper Networks, Inc.

Related Documentation

- Example: Improving Commit Time When Using Configuration Groups on page 223
- *persist-groups-inheritance*

Example: Improving Commit Time When Using Configuration Groups

This example shows how to use the **persist-groups-inheritance** configuration statement to improve commit time performance when committing configurations that make use of many configuration groups that are created using wildcards.

- Requirements on page 223
- Overview on page 223
- Configuration on page 223
- Verification on page 224

Requirements

This example uses the following hardware and software components:

- One Juniper Networks M Series, MX Series, or T Series router that uses a number of configuration groups created with wildcards.
- Junos OS Release 13.2 or later.

Overview

When committing a configuration that uses configuration groups, at the time of commit, all of the inheritance paths of the configuration groups must be fully expanded into memory to apply the configurations as intended. This can negatively impact commit performance if there are many configuration groups and they are configured using wildcards.

To improve commit performance, you can configure **persist-groups-inheritance** at the **[edit system commit]** hierarchy level. Configuring this option causes the configuration groups to be expanded into the database instead of into the process memory at commit time.

Configuration

Configuring Persist Groups Inheritance

Step-by-Step Procedure

To configure **persist-groups-inheritance**:

1. Set the **persist-groups-inheritance** option.

   ```
   [edit system commit]
   user@host# set persist-groups-inheritance
   ```

2. Commit the configuration.

   ```
   [edit system commit]
   user@host# commit
   ```

Copyright © 2014, Juniper Networks, Inc.

Verification

Verifying the Configuration

Purpose Verify that **persist-groups-inheritance** is configured.

Action To confirm the configuration, use the **show system commit** command.

```
[edit ]
user@host# show system commit
persist-groups-inheritance
```

Related Documentation

- Improving Commit Time When Using Configuration Groups on page 222
- *persist-groups-inheritance*

Using Junos OS Defaults Groups

Junos OS provides a hidden and immutable configuration group called **junos-defaults** that is automatically applied to the configuration of your router. The **junos-defaults** group contains preconfigured statements that contain predefined values for common applications. Some of the statements must be referenced to take effect, such as definitions for applications (for example, FTP or telnet settings). Other statements are applied automatically, such as terminal settings.

NOTE: Many identifiers included in the **junos-defaults** configuration group begin with the name **junos-**. Because identifiers beginning with the name **junos-** are reserved for use by Juniper Networks, you cannot define any configuration objects using this name.

You cannot include **junos-defaults** as a configuration group name in an **apply-groups** statement.

To view the full set of available preset statements from the Junos defaults group, issue the **show groups junos-defaults** configuration mode command at the top level of the configuration. The following example displays a partial list of Junos defaults groups:

```
user@host# show groups junos-defaults
# Make vt100 the default for the console port
system {
    ports {
        console type vt100;
    }
}
applications {
    # File Transfer Protocol
    application junos-ftp {
        application-protocol ftp;
        protocol tcp;
```


Copyright © 2014, Juniper Networks, Inc.

```
    destination-port 21;
  }
  # Trivial File Transfer Protocol
  application junos-tftp {
    application-protocol tftp;
    protocol udp;
    destination-port 69;
  }
  # RPC port mapper on TCP
  application junos-rpc-portmap-tcp {
    application-protocol rpc-portmap;
    protocol tcp;
    destination-port 111;
  }
  # RPC port mapper on UDP
}
```

To reference statements available from the **junos-defaults** group, include the selected **junos-** ***default-name*** statement at the applicable hierarchy level.

Related Documentation

- Creating a Junos OS Configuration Group on page 199
- Example: Referencing the Preset Statement From the Junos OS defaults Group on page 218
- Example: Viewing Default Statements That Have Been Applied to the Configuration on page 219

Set Up Routing Engine Configuration Groups

In a router with two Routing Engines, one configuration should be shared between both Routing Engines. This ensures that both Routing Engine configurations are identical. Within this configuration, create two Routing Engine groups, one for each Routing Engine. Within these groups, you specify the Routing Engine–specific parameters.

For more information about creating configuration groups, see *CLI User Guide*.

For more information about the initial configuration for redundant Routing Engine systems and the re0 group, see *Junos OS High Availability Library for Routing Devices*.

1. Create the configuration group **re0**. The **re0** group is a special group designator that is only used by **RE0** in a redundant routing platform.

   ```
   [edit]
   root# set groups re0
   ```

2. Navigate to the **groups re0** level of the configuration hierarchy.

   ```
   [edit]
   root#  edit groups re0
   ```

3. Specify the router hostname.

   ```
   [edit groups re0]
   root# set system host-name host-name
   ```

Copyright © 2014, Juniper Networks, Inc.

NOTE: The hostname specified in the router configuration is not used by the DNS server to resolve to the correct IP address. This hostname is used to display the name of the Routing Engine in the CLI. For example, the hostname appears at the command-line prompt when the user is logged in to the CLI:

user-name@*host-name*>

4. Configure the IP address and prefix length for the router Ethernet interface.

   - For all devices *except* the TX Matrix Plus router, T1600 or T4000 routers in a routing matrix, and PTX Series Packet Transport Routers:

     ```
     [edit]
     root@# set interfaces fxp0 unit 0 family inet address address/prefix-length
     ```

   - For TX Matrix Plus router, and T1600 or T4000 routers in a routing matrix only, and PTX Series Packet Transport Routers:

     ```
     [edit]
     root@# set interfaces em0 unit 0 family inet address address/prefix-length
     ```

     To use **em0** as an out-of-band management Ethernet interface, you must configure its logical port, **em0.0**, with a valid IP address.

   - For a T1600 standalone router (not connected to a TX Matrix Plus router and not in a routing matrix):

     ```
     [edit]
     root@# set interfaces fxp0 unit 0 family inet address address/prefix-length
     ```

5. Return to the top level of the hierarchy.

   ```
   [edit groups re0]
   root# top
   ```

6. Create the configuration group **re1**.

   ```
   [edit]
   root# set groups re1
   ```

7. Navigate to the **groups re1** level of the configuration hierarchy.

   ```
   [edit]
   root# edit groups re1
   ```

8. Specify the router hostname.

   ```
   [edit groups re1]
   root# set system host-name host-name
   ```

9. Configure the IP address and prefix length for the router Ethernet interface.

   - For all devices *except* the TX Matrix Plus router, T1600 or T4000 routers in a routing matrix, and PTX Series Packet Transport Routers:

     ```
     [edit]
     root@# set interfaces fxp0 unit 0 family inet address address/prefix-length
     ```


Copyright © 2014, Juniper Networks, Inc.

- For TX Matrix Plus router, and T1600 or T4000 routers in a routing matrix only:

  ```
  [edit]
  root@# set interfaces em0 unit 0 family inet address address/prefix-length
  ```

  To use **em0** as an out-of-band management Ethernet interface, you must configure its logical port, **em0.0**, with a valid IP address.

- For a T1600 standalone router (not connected to a TX Matrix Plus router, and not in a routing matrix):

  ```
  [edit]
  root@# set interfaces fxp0 unit 0 family inet address address/prefix-length
  ```

10. Return to the top level of the hierarchy.

    ```
    [edit groups re0]
    root# top
    ```

11. Specify the group application order.

    ```
    [edit]
    root# set apply-groups [ re0 re1 ]
    ```

Copyright © 2014, Juniper Networks, Inc.

Copyright © 2014, Juniper Networks, Inc.

CHAPTER 11

Controlling the CLI Environment

- Controlling the Junos OS CLI Environment on page 229
- Setting the Junos OS CLI Screen Length and Width on page 231
- Example: Controlling the CLI Environment on page 232
- Example: Enabling Configuration Breadcrumbs on page 233

Controlling the Junos OS CLI Environment

In operational mode, you can control the Junos OS command-line interface (CLI) environment. For example, you can specify the number of lines that are displayed on the screen or your terminal type. The following output lists the options that you can use to control the CLI environment:

```
user@host>set cli ?
Possible completions:
  complete-on-space    Set whether typing space completes current word
  directory            Set working directory
  idle-timeout         Set maximum idle time before login session ends
  logical-system       Set default logical system
  prompt               Set CLI command prompt string
  restart-on-upgrade   Set whether CLI prompts to restart after software upgrade

  screen-length        Set number of lines on screen
  screen-width         Set number of characters on a line
  terminal             Set terminal type
  timestamp            Timestamp CLI output
```

NOTE: When you use SSH to log in to the router or log in from the console when its terminal type is already configured (as described in the *Junos OS Administration Library for Routing Devices*), your terminal type, screen length, and screen width are already set.

This chapter discusses the following topics:

- Setting the Terminal Type on page 230
- Setting the CLI Prompt on page 230
- Setting the CLI Directory on page 230
- Setting the CLI Timestamp on page 230

Copyright © 2014, Juniper Networks, Inc.

- Setting the Idle Timeout on page 230
- Setting the CLI to Prompt After a Software Upgrade on page 230
- Setting Command Completion on page 231
- Displaying CLI Settings on page 231

Setting the Terminal Type

To set the terminal type, use the **set cli terminal** command:

```
user@host> set cli terminal terminal-type
```

The terminal type can be one of the following: **ansi**, **vt100**, **small-xterm**, or **xterm**.

Setting the CLI Prompt

The default CLI prompt is **user@host>**. To change this prompt, use the **set cli prompt** command. If the prompt string contains spaces, enclose the string in quotation marks (" ").

```
user@host> set cli prompt string
```

Setting the CLI Directory

To the set the current working directory, use the **set cli directory** command:

```
user@host> set cli directory directory
```

directory is the pathname of working directory.

Setting the CLI Timestamp

By default, CLI output does not include a timestamp. To include a timestamp in CLI output, use the **set cli timestamp** command:

```
user@host> set cli timestamp [format time-date-format | disable]
```

If you do not specify a timestamp format, the default format is ***Mmm dd hh:mm:ss*** (for example, Feb 08 17:20:49). Enclose the format in single quotation marks (‘).

Setting the Idle Timeout

By default, an individual CLI session never times out after extended times, unless the **idle-timeout** statement has been included in the user’s login class configuration. To set the maximum time an individual session can be idle before the user is logged off the router, use the **set cli idle-timeout** command:

```
user@host> set cli idle-timeout timeout
```

timeout can be **0** through **100,000** minutes. Setting *timeout* to **0** disables the timeout.

Setting the CLI to Prompt After a Software Upgrade

By default, the CLI prompts you to restart after a software upgrade. To disable the prompt for an individual session, use the **set cli restart-on-upgrade off** command:

```
user@host>set cli restart-on-upgrade off
```


Copyright © 2014, Juniper Networks, Inc.

To reenable the prompt, use the **set cli restart-on-upgrade on** command:

```
user@host> set cli restart-on-upgrade on
```

Setting Command Completion

By default, you can press Tab or the Spacebar to have the CLI complete a command.

To have the CLI allow only a tab to complete a command, use the **set cli complete-on-space off** command:

```
user@host> set cli complete-on-space off
Disabling complete-on-space
user@host>
```

To reenable the use of both spaces and tabs for command completion, use the **set cli complete-on-space on** command:

```
user@host> set cli complete-on-space on
Enabling complete-on-space
user@host>
```

Displaying CLI Settings

To display the current CLI settings, use the **show cli** command:

```
user@host> show cli
CLI screen length set to 24
CLI screen width set to 80
CLI complete-on-space set to on
```

Related Documentation

- Example: Controlling the CLI Environment on page 232

Setting the Junos OS CLI Screen Length and Width

You can set the Junos OS command-line interface (CLI) screen length and width according to your specific requirements. This topic contains the following sections:

- Setting the Screen Length on page 231
- Setting the Screen Width on page 232
- Understanding the Screen Length and Width Settings on page 232

Setting the Screen Length

The default CLI screen length is 24 lines. To change the length, use the **set cli screen-length** command:

```
user@host> set cli screen-length length
```

Setting the screen length to 0 lines disables the display of output one screen at a time. Disabling this UNIX **more**-type interface can be useful when you are issuing CLI commands from scripts.

Copyright © 2014, Juniper Networks, Inc.

Setting the Screen Width

The default CLI screen width is 80 characters. To change the width, use the **set cli screen-width** command:

```
user@host> set cli screen-width width
```

NOTE: In Junos OS Release 13.3 and later, the value of *width* is 0 or in the range of 40 through 1024. In earlier Junos releases, the range is 0 through 1024.

Understanding the Screen Length and Width Settings

The **cli screen-length** and **cli screen-width** settings in combination with each other and the size of the telnet or console window determine the extent of output displayed before each **–more–** prompt appears.

The following examples explain how the **cli screen-length** and **cli screen-width** values determine the appearance of the output:

- When the CLI screen width is set to the default value (80 characters) and the cli scree length to 10 lines, the **--more--** prompt appears on the tenth line of the output.

NOTE: If you have inadvertently set the CLI screen width to a lower value that does not allow you to see the commands that you are typing, reset the CLI screen width with a higher value by entering the **set cli screen-width** command.

TIP: If you are not able to see the command that you are entering, type the command in a text editor and copy it at the command prompt.

Related Documentation

- Example: Controlling the CLI Environment on page 232
- Controlling the Junos OS CLI Environment on page 229

Example: Controlling the CLI Environment

The following example shows you how to change the default CLI environment:

```
user@host> set cli screen-length 66
Screen length set to 66
user@host> set cli screen-width 40
Screen width set to 40
user@host> set cli prompt "router1-san-jose > "
router1-san-jose > show cli
CLI complete-on-space set to on
```


Copyright © 2014, Juniper Networks, Inc.

```
CLI idle-timeout disabled
CLI restart-on-upgrade set to on
CLI screen length set to 66
CLI screen width set to 40
CLI terminal is 'xterm'
router1-san-jose >
```

Related Documentation

- Setting the Junos OS CLI Screen Length and Width on page 231
- Controlling the Junos OS CLI Environment on page 229

Example: Enabling Configuration Breadcrumbs

The output of **show configuration** operational mode command and **show** configuration mode commands can be configured to display configuration breadcrumbs that indicate the exact location in the hierarchy of the output being viewed.

Before enabling the configuration breadcrumbs feature, check the output of the **show configuration** command.

```
user@host> show configuration

...
                }
            }
        }
    }
  }
  fe-4/1/2 {
      description "FA4/1/2: mxxj1-mr6 (64.12.137.160/27) (T=bblan, bbmail,
bbowmtc)";
      unit 0 {
          family inet {
              filter {
                  output 151mj;
              }
              address 64.12.137.187/27 {
                  vrrp-group 1 {
                      virtual-address 64.12.137.189;
---(more 18%)--------------------------------------------------------
```

In the output, there is no clear indication about the section of the configuration being viewed.

To enable the configuration breadcrumbs feature:

1. Define a class at the **[edit system login]** hierarchy level.

   ```
   [edit system login]
   user@host# set class breadclass idle-timeout 10
   ```

2. Add a user to the defined login class to enable the breadcrumbs output view when this user enters the **show configuration** operational mode command.

   ```
   [edit system login user user1]
   user@host# set class breadclass
   ```

Copyright © 2014, Juniper Networks, Inc.

3. Configure the **configuration-breadcrumbs** statement at the **[edit system login class *<class name>*]** hierarchy level.

   ```
   [edit system login class breadclass]
   user@host# set configuration-breadcrumbs
   ```

4. Confirm the configuration.

   ```
   [edit]
   user@host# commit
   ```

   On enabling configuration breadcrumbs in the CLI, User1 (the user added to the login class) can verify the feature in the output by entering the **show configuration** command.

   ```
   user1@host> show configuration

   ...
                   }
                }
             }
          }
     }
     fe-4/1/2 {
          description "FA4/1/2: mxxj1-mr6 (64.12.137.160/27) (T=bblan, bbmail,
   bbowmtc)";
          unit 0 {
              family inet {
                  filter {
                      output 151mj;
                  }
                  address 64.12.137.187/27 {
                      vrrp-group 1 {
                          virtual-address 64.12.137.189;
   ---(more 18%)---[groups main interfaces fe-4/1/2 unit 0 family inet address
   64.12.137.187/27 vrrp-group 1]---
   ```

   The new output indicates the exact location of the configuration hierarchy being viewed. User1 is currently viewing the interface configuration of a group.

   NOTE: If you are enabling configuration breadcrumbs for your own user account, you should log out and log in again to see the changes.

Related Documentation

- *class*
- configuration-breadcrumbs on page 245

Copyright © 2014, Juniper Networks, Inc.

CHAPTER 12

Junos OS Configuration Statements and Commands

- apply-groups on page 236
- apply-groups-except on page 237
- activate
- annotate
- commit
- commit-interval (Batch Commits) on page 245
- configuration-breadcrumbs on page 245
- copy
- days-to-keep-error-logs (Batch Commits) on page 246
- deactivate
- delete
- edit
- exit
- groups on page 251
- help
- insert
- load
- maximum-aggregate-pool (Batch Commits) on page 256
- maximum-entries (Batch Commits) on page 257
- protect
- quit
- rename
- replace
- rollback
- run
- save

Copyright © 2014, Juniper Networks, Inc.

- server (Batch Commits) on page 265
- set
- show
- show configuration
- show | display inheritance
- show | display omit
- show | display set
- show | display set relative
- show groups junos-defaults
- status
- top
- traceoptions (Batch Commits) on page 278
- unprotect
- up
- update
- when on page 282
- wildcard delete

apply-groups

Syntax apply-groups [*group-names*];

Hierarchy Level All hierarchy levels

Release Information Statement introduced before Junos OS Release 7.4.

Description Apply a configuration group to a specific hierarchy level in a configuration, to have a configuration inherit the statements in the configuration group.

You can specify more than one group name. You must list them in order of inheritance priority. The configuration data in the first group takes priority over the data in subsequent groups.

Options ***group-names***—One or more names specified in the **groups** statement.

Required Privilege Level configure—To enter configuration mode, but other required privilege levels depend on where the statement is located in the configuration hierarchy.

Related Documentation

- Applying a Junos OS Configuration Group on page 201
- **groups on page 251**

Copyright © 2014, Juniper Networks, Inc.

apply-groups-except

Syntax apply-groups-except [*group-names*];

Hierarchy Level All hierarchy levels except the top level

Release Information Statement introduced before Junos OS Release 7.4.

Description Disable inheritance of a configuration group.

Options ***group-names***—One or more names specified in the **groups** statement.

Required Privilege Level configure—To enter configuration mode, but other required privilege levels depend on where the statement is located in the configuration hierarchy.

Related Documentation

- groups on page 251
- Disabling Inheritance of a Junos OS Configuration Group on page 204

Copyright © 2014, Juniper Networks, Inc.

activate

Syntax activate (*statement* | *identifier*)

Release Information Command introduced before Junos OS Release 7.4.

Description Remove the **inactive:** tag from a statement, effectively adding the statement or identifier back to the configuration. Statements or identifiers that have been activated take effect when you next issue the **commit** command.

Options ***identifier***—Identifier from which you are removing the **inactive** tag. It must be an identifier at the current hierarchy level.

statement—Statement from which you are removing the **inactive** tag. It must be a statement at the current hierarchy level.

Required Privilege Level configure—To enter configuration mode, but other required privilege levels depend on where the statement is located in the configuration hierarchy.

Related Documentation

- deactivate on page 247
- Deactivating and Reactivating Statements and Identifiers in a Junos OS Configuration on page 82

Copyright © 2014, Juniper Networks, Inc.

annotate

Syntax annotate ***statement*** "***comment-string***"

Release Information Command introduced before Junos OS Release 7.4.

Description Add comments to a configuration. You can add comments only at the current hierarchy level.

Any comments you add appear only when you view the configuration by entering the show command in configuration mode or the **show configuration** command in operational mode.

NOTE: The Junos OS supports annotation up to the last level in the configuration hierarchy, including onliners. However, annotation of parts (child statements or identifiers within a oneliner) of the onliner is not supported. For example, in the following sample configuration hierarchy, annotation is supported up to the onliner level 1 , but not supported for the metric child statement and its attribute *10*:

```
[edit protocols]
   isis {
      interface ge-0/0/0.0 {
         level 1 metric 10;
      }
   }
}
```

Options ***comment-string***—Text of the comment. You must enclose it in quotation marks. In the comment string, you can include the comment delimiters **/* */** or **#**. If you do not specify any, the comment string is enclosed with the **/* */** comment delimiters. If a comment for the specified ***statement*** already exists, it is deleted and replaced with the new comment.

statement—Statement to which you are attaching the comment.

Required Privilege Level configure—To enter configuration mode, but other required privilege levels depend on where the statement is located in the configuration hierarchy.

Related Documentation

- Adding Comments in a Junos OS Configuration on page 84

Copyright © 2014, Juniper Networks, Inc.

commit

Syntax commit <<*at* <"string">> <and-quit> <check> <comment <"*comment-string*">> <confirmed> <display detail> <**fast-synchronize**> <*minutes*> <synchronize <force> <scripts>>

Release Information Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 11.1 for the QFX Series.
Option **fast-synchronize** added in Junos OS Release 12.2.
Option **synchronize scripts** introduced in Junos OS Release 13.2.

Description Commit the set of changes to the database and cause the changes to take operational effect.

NOTE: The **fast-synchronize** option is not supported in a QFX Series Virtual Chassis.

NOTE: Beginning in Junos OS 12.3, it is possible that FPCs brought offline using the **request chassis fpc slot** *fpc-slot* **offline** operational-mode CLI command can come online during a configuration commit or power-supply replacement procedure. As an alternative, use the **set fpc** *fpc-slot* **power off** configuration-mode command at the **[edit chassis]** hierarchy level to ensure that the FPCs remain offline.

Options **at <"string">**—(Optional) Save software configuration changes and activate the configuration at a future time, or upon reboot.

string is **reboot** or the future time to activate the configuration changes. Enclose the ***string*** value (including **reboot**) in quotation marks (“ ”). You can specify time in two formats:

- A time value in the form ***hh:mm***[**:*ss***] (hours, minutes, and optionally seconds)—Commit the configuration at the specified time, which must be in the future but before 11:59:59 PM on the day the **commit at** configuration command is issued. Use 24-hour time for the ***hh*** value; for example, **04:30:00** is 4:30:00 AM, and **20:00** is 8:00 PM. The time is interpreted with respect to the clock and time zone settings on the router.
- A date and time value in the form ***yyyy-mm-dd hh:mm***[**:*ss***] (year, month, date, hours, minutes, and, optionally, seconds)—Commit the configuration at the specified day and time, which must be after the **commit at** command is issued. Use 24-hour time for the ***hh*** value. For example, **2003-08-21 12:30:00** is 12:30 PM on August 21, 2003. The time is interpreted with respect to the clock and time zone settings on the router.

For example, **commit at “18:00:00"**. For date and time, include both values in the same set of quotation marks. For example, **commit at "2005-03-10 14:00:00"**.

Copyright © 2014, Juniper Networks, Inc.

A *commit check* is performed when you issue the **commit at** configuration mode command. If the result of the check is successful, then the current user is logged out of configuration mode, and the configuration data is left in a read-only state. No other commit can be performed until the scheduled commit is completed.

NOTE: If Junos OS fails before the configuration changes become active, all configuration changes are lost.

You cannot enter the **commit at** configuration command when there is a pending reboot.

You cannot enter the **request system reboot** command once you schedule a commit operation for a specific time in the future.

You cannot commit a configuration when a scheduled commit is pending. For information about how to use the **clear** command to cancel a scheduled configuration, see the CLI Explorer.

and-quit—(Optional) Commit the configuration and, if the configuration contains no errors and the commit succeeds, exit from configuration mode.

check—(Optional) Verify the syntax of the configuration, but do not activate it.

comment <"***comment-string***">—(Optional) Add a comment that describes the committed configuration. The comment can be as long as 512 bytes and must be typed on a single line. You cannot include a comment with the **commit check** command. Enclose ***comment-string*** in quotation marks (" "). For example, **commit comment "Includes changes recommended by SW Lab".**

confirmed <*minutes*>—(Optional) Require that the commit be confirmed within the specified amount of time. To confirm a commit, enter either a **commit** or **commit check** command. If the commit is not confirmed within the time limit, the configuration rolls back automatically to the precommit configuration and a broadcast message is sent to all logged-in users. To show when a rollback is scheduled, enter the **show system commit** command. The allowed range is **1** through **65,535** minutes, and the default is **10** minutes.

In Junos OS Release 11.4 and later, you can also use the **commit confirmed** command in the **[edit private]** configuration mode.

display detail—(Optional) Monitors the commit process.

NOTE: In Junos OS Release 10.4 and later, if the number of commit details or messages exceeds a page when used with the **| display detail** pipe option, the **more** pagination option on the screen is no longer available. Instead, the messages roll up on the screen by default, just like using the **commit** command with the **| no more** pipe option.

Copyright © 2014, Juniper Networks, Inc.

fast-synchronize—(Optional) Configure the commits to run in parallel on both the master and backup Routing Engines to reduce the time taken for commit synchronization.

NOTE: The **fast-synchronize** statement is not supported on QFX Series devices when used in a Virtual Chassis.

synchronize <force> <scripts>—(Optional) If your router has two Routing Engines, you can manually direct one Routing Engine to synchronize its configuration with the other by issuing the **commit synchronize** command. The Routing Engine on which you execute this command (request Routing Engine) copies and loads its candidate configuration to the other (responding Routing Engine). Both Routing Engines then perform a syntax check on the candidate configuration file being committed. If no errors are found, the configuration is activated and becomes the current operational configuration on both Routing Engines. The **commit synchronize** command does not work if the responding Routing Engine has uncommitted configuration changes. However, you can enforce commit synchronization on the Routing Engines by using the **force** option. When you issue the **commit synchronize** command with the **force** option from one Routing Engine, the configuration sessions on the other Routing Engine are terminated and its configuration synchronized with that on the Routing Engine from which you issued the command.

When you issue the **commit synchronize** command with the **scripts** option, the device synchronizes all commit, event, lib, and op scripts from the requesting Routing Engine to the responding Routing Engine and also commits and synchronizes the configuration. If the commit check operation fails for the requesting Routing Engine, the process stops, and the scripts are not copied to the responding Routing Engine. If the commit check or commit operation fails for the responding Routing Engine, the scripts are still synchronized, since the synchronization occurs prior to the commit check operation on the responding Routing Engine.

If the **load-scripts-from-flash** statement is configured for the requesting Routing Engine, the device synchronizes the scripts from flash memory on the requesting Routing Engine to flash memory on the responding Routing Engine. Otherwise, the device synchronizes the scripts from the hard disk on the requesting Routing Engine to the hard disk on the responding Routing Engine. The device synchronizes all scripts regardless of whether they are enabled in the configuration or have been updated since the last synchronization.

NOTE: When you issue the **commit synchronize** command, you must use the **apply-groups re0** and **re1** commands. For information about how to use groups, see "Disabling Inheritance of a Junos OS Configuration Group" on page 204.

The responding Routing Engine must use Junos OS Release 5.0 or later.

Required Privilege Level configure—To enter configuration mode.

Copyright © 2014, Juniper Networks, Inc.

NOTE: If you are using Junos OS in a Common Criteria environment, system log messages are created whenever a secret attribute is changed (for example, password changes or changes to the RADIUS shared secret). These changes are logged during the following configuration load operations:

```
load merge
load replace
load override
load update
```

For more information, see the *Secure Configuration Guide for Common Criteria and Junos-FIPS*

Related Documentation

- Verifying a Junos OS Configuration on page 95, Committing a Junos OS Configuration on page 98
- Scheduling a Junos OS Commit Operation on page 103
- Deactivating and Reactivating Statements and Identifiers in a Junos OS Configuration on page 82
- Monitoring the Junos OS Commit Process on page 104
- Adding a Comment to Describe the Committed Configuration on page 105

Sample Output

commit | display detail

```
user@host> commit | display detail
--------------
2011-08-24 01:08:08.00691 PDT: begin creating snapshots
2011-08-24 01:08:09.00210 PDT: end creating snapshots
2011-08-24 01:08:09.00211 PDT: begin preparing metadata
2011-08-24 01:08:09.00228 PDT: end preparing metadata
2011-08-24 01:08:09.00229 PDT: begin computing dcf root changes
2011-08-24 01:08:09.00236 PDT: end computing dcf root changes
2011-08-24 01:08:09.00244 PDT: begin computing additions
2011-08-24 01:08:09.00251 PDT: end computing additions
2011-08-24 01:08:09.00251 PDT: begin local object validation
2011-08-24 01:08:09.00251 PDT: end local object validation
2011-08-24 01:08:09.00252 PDT: begin update instances
2011-08-24 01:08:09.00252 PDT: end update instances
2011-08-24 01:08:09.00252 PDT: begin adjust metadata
2011-08-24 01:08:09.00252 PDT: end adjust metadata
2011-08-24 01:08:09.00253 PDT: begin validate metadata
2011-08-24 01:08:09.00253 PDT: end validate metadata
2011-08-24 01:08:09.00253 PDT: begin adjust allocations
2011-08-24 01:08:09.00254 PDT: end adjust allocations
2011-08-24 01:08:09.00254 PDT: begin adjust dependencies
2011-08-24 01:08:09.00254 PDT: end adjust dependencies
2011-08-24 01:08:09.00255 PDT: begin instance validation
2011-08-24 01:08:09.00255 PDT: end instance validation
2011-08-24 01:08:09.00255 PDT: begin opening all sessions eagerly
2011-08-24 01:08:09.00277 PDT: begin request #1 [login]
```

Copyright © 2014, Juniper Networks, Inc.

```
2011-08-24 01:08:09.00278 PDT: end request #1 [login]
2011-08-24 01:08:09.00325 PDT: begin processing globals
2011-08-24 01:08:09.00330 PDT: begin waiting for stamp check
(qfabric-default---node0)
2011-08-24 01:08:09.00334 PDT: end reply #1 [login]
2011-08-24 01:08:09.00351 PDT: end reply #1 [login]
2011-08-24 01:08:09.00451 PDT: begin request #2 [open]
2011-08-24 01:08:09.00451 PDT: end request #2 [open]
2011-08-24 01:08:09.00451 PDT: begin request #3 [get commit history]
2011-08-24 01:08:09.00452 PDT: end request #3 [get commit history]
2011-08-24 01:08:09.00452 PDT: begin request #4 [load]
2011-08-24 01:08:09.00453 PDT: end request #4 [load]
2011-08-24 01:08:09.00453 PDT: begin request #5 [load]
2011-08-24 01:08:09.00454 PDT: begin reply #2 [open]
2011-08-24 01:08:09.00456 PDT: end reply #2 [open]
2011-08-24 01:08:09.00457 PDT: begin reply #3 [get commit history]
2011-08-24 01:08:09.00475 PDT: end reply #3 [get commit history]
2011-08-24 01:08:09.00476 PDT: begin reply #4 [load]
2011-08-24 01:08:09.00499 PDT: begin reply #5 [load]
2011-08-24 01:08:09.00501 PDT: end waiting for stamp check
(qfabric-default---node0)
2011-08-24 01:08:09.00501 PDT: begin waiting for open (qfabric-default---node0)
2011-08-24 01:08:09.00502 PDT: end waiting for open (qfabric-default---node0)
2011-08-24 01:08:09.00504 PDT: end processing globals
2011-08-24 01:08:09.00617 PDT: end request #5 [load]
2011-08-24 01:08:09.00617 PDT: begin request #6 [check]
2011-08-24 01:08:09.00617 PDT: end request #6 [check]
2011-08-24 01:08:09.00619 PDT: end reply #5 [load]
2011-08-24 01:08:09.00619 PDT: begin reply #6 [check]
2011-08-24 01:08:09.00730 PDT: end session
2011-08-24 01:08:09.00752 PDT: end request #5 [load]
2011-08-24 01:08:09.00754 PDT: begin request #6 [check]
2011-08-24 01:08:09.00755 PDT: end request #6 [check]
2011-08-24 01:08:09.00881 PDT: end request #5 [load]
2011-08-24 01:08:09.00961 PDT: begin commit to devices
2011-08-24 01:08:10.00668 PDT: begin request #8 [get commit history]
2011-08-24 01:08:10.00669 PDT: end request #8 [get commit history]
2011-08-24 01:08:10.00721 PDT: end session
2011-08-24 01:08:10.00727 PDT: end commit to devices
2011-08-24 01:08:10.00733 PDT: begin committing metadata
2011-08-24 01:08:10.00772 PDT: end committing metadata
2011-08-24 01:08:10.00772 PDT: begin calling commit callbacks
2011-08-24 01:08:10.00773 PDT: end calling commit callbacks
commit complete
```


Copyright © 2014, Juniper Networks, Inc.

commit-interval (Batch Commits)

Syntax commit-interval *number-of-seconds-between-commits*;

Hierarchy Level [edit system commit server],
[edit system commit synchronize server]

Release Information Statement introduced in Junos OS Release 12.1.

Description For Junos OS batch commits, specify the time interval (in seconds) between two commit operations.

Options ***number-of-seconds-between-commits***—Time interval (in seconds) between two commit operations.

Range: 1 through 30 seconds.

Default: 5 seconds.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Batch Commit Server Properties on page 107

configuration-breadcrumbs

Syntax configuration-breadcrumbs;

Hierarchy Level [edit system login class]

Release Information Statement introduced in Junos OS Release 12.2.

Description Enable the configuration breadcrumbs view in the CLI to display the location in the configuration hierarchy.

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

Related Documentation

- Example: Enabling Configuration Breadcrumbs on page 233
- *Defining Junos OS Login Classes*
- *class*
- *login*

Copyright © 2014, Juniper Networks, Inc.

copy

Syntax copy *existing-statement* to *new-statement*

Release Information Command introduced before Junos OS Release 7.4.

Description Make a copy of an existing statement in the configuration.

Options *existing-statement*—Statement to copy.

new-statement—Copy of the statement.

Required Privilege Level configure—To enter configuration mode, but other required privilege levels depend on where the statement is located in the configuration hierarchy.

Related Documentation
- Copying a Junos OS Statement in the Configuration on page 73

days-to-keep-error-logs (Batch Commits)

Syntax days-to-keep-error-logs *days-to-keep-error-log-entries*;

Hierarchy Level [edit system commit server],
[edit system commit synchronize server]

Release Information Statement introduced in Junos OS Release 12.1.

Description For Junos OS batch commits, specify the number of days to keep the error logs.

Options *days-to-keep-error-log-entries*—Number of days to keep the error logs.

Range: 1 through 366 days

Default: 1 day

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring Batch Commit Server Properties on page 107

Copyright © 2014, Juniper Networks, Inc.

deactivate

Syntax deactivate (*statement* | *identifier*)

Release Information Command introduced before Junos OS Release 7.4.

Description Add the **inactive:** tag to a statement, effectively commenting out the statement or identifier from the configuration. Statements or identifiers marked as inactive do not take effect when you issue the **commit** command.

Options *identifier*—Identifier to which you are adding the **inactive:** tag. It must be an identifier at the current hierarchy level.

statement—Statement to which you are adding the **inactive:** tag. It must be a statement at the current hierarchy level.

Required Privilege Level configure—To enter configuration mode, but other required privilege levels depend on where the statement is located in the configuration hierarchy.

Related Documentation

- activate on page 238
- delete on page 248
- Deactivating and Reactivating Statements and Identifiers in a Junos OS Configuration on page 82.

Copyright © 2014, Juniper Networks, Inc.

delete

Syntax delete <*statement-path*> <*identifier*>

Release Information Command introduced before Junos OS Release 7.4.

Description Delete a statement or identifier. All subordinate statements and identifiers contained within the specified statement path are deleted with it.

Deleting a statement or an identifier effectively "unconfigures" or disables the functionality associated with that statement or identifier.

If you do not specify ***statement-path*** or ***identifier***, the entire hierarchy, starting at the current hierarchy level, is removed.

Options ***statement-path***—(Optional) Path to an existing statement or identifier. Include this if the statement or identifier to be deleted is not at the current hierarchy level.

identifier—(Optional) Name of the statement or identifier to delete.

Required Privilege Level configure—To enter configuration mode, but other required privilege levels depend on where the statement is located in the configuration hierarchy.

Related Documentation

- deactivate on page 247
- Deleting a Statement from a Junos OS Configuration on page 71

Copyright © 2014, Juniper Networks, Inc.

edit

Syntax edit *statement-path*

Release Information Command introduced before Junos OS Release 7.4.

Description Move inside the specified statement hierarchy. If the statement does not exist, it is created.

You cannot use the **edit** command to change the value of identifiers. You must use the **set** command.

Options *statement-path*—Path to the statement.

Required Privilege Level configure—To enter configuration mode, but other required privilege levels depend on where the statement is located in the configuration hierarchy.

Related Documentation

- set on page 266
- Displaying the Current Junos OS Configuration on page 87

Copyright © 2014, Juniper Networks, Inc.

exit

Syntax exit <configuration-mode>

Release Information Command introduced before Junos OS Release 7.4.

Description Exit the current level of the statement hierarchy, returning to the level prior to the last **edit** command, or exit from configuration mode. The **quit** and **exit** commands are synonyms.

Options none—Return to the previous edit level. If you are at the top of the statement hierarchy, exit configuration mode.

configuration-mode—(Optional) Exit from configuration mode.

Required Privilege Level configure—To enter configuration mode, but other required privilege levels depend on where the statement is located in the configuration hierarchy.

Related Documentation

- top on page 277
- up on page 280
- Displaying the Current Junos OS Configuration on page 87

Copyright © 2014, Juniper Networks, Inc.

groups

Syntax

```
groups {
  group-name {
    configuration-data;
    when {
      chassis chassis-id;
      member member-id;
      model model-id;
      node node-id;
      routing-engine routing-engine-id;
      time <start-time> [to <end-time>];
    }
      conditional-data;
  }
  lccn-re0 {
    configuration-data;
  }
  lccn-re1 {
    configuration-data;
  }
}
```

Hierarchy Level [edit]

Release Information Statement introduced before Junos OS Release 7.4.

Description Create a configuration group.

Options —

group-name——Name of the configuration group. To configure multiple groups, specify more than one ***group-name***.

configuration-data—The configuration statements that are to be applied elsewhere in the configuration with the **apply-groups** statement, to have the target configuration inherit the statements in the group.

when *conditional-data*—Option introduced in Junos 11.3. The conditional statements that are to be applied when this configuration group is applied.

On routers that support multiple Routing Engines, you can also specify two special group names:

re0—Configuration statements that are to be applied to the Routing Engine in slot 0.

re1—Configuration statements that are to be applied to the Routing Engine in slot 1.

The configuration specified in group **re0** is applied only if the current Routing Engine is in slot 0; likewise, the configuration specified in group **re1** is applied only if the current Routing Engine is in slot 1. Therefore, both Routing Engines can use the same configuration file, each using only the configuration statements that apply to it. Each **re0** or **re1** group contains at a minimum the configuration for the hostname and the

Copyright © 2014, Juniper Networks, Inc.

management interface (**fxp0**). If each Routing Engine uses a different management interface, the group also should contain the configuration for the backup router and static routes.

(Routing matrix only) The TX Matrix router supports group names for the Routing Engines in each connected T640 router in the following formats:

NOTE: The management Ethernet interface used for the TX Matrix Plus router, T1600 routers in a routing matrix, and PTX Series Packet Transport Routers, is em0. Junos OS automatically creates the router's management Ethernet interface, em0.

- **lcc*n*-re0**—Configuration statements applied to the Routing Engine in slot 0 of the specified T640 router that is connected to a TX Matrix router.
- **lcc*n*-re1**—Configuration statements applied to the specified to the Routing Engine in slot 1 of the specified T640 router that is connected to a TX Matrix router.

  n identifies the T640 router and can be from 0 through 3.

The remaining statements are explained separately.

Required Privilege Level

configure—To enter configuration mode.

Related Documentation

- Creating a Junos OS Configuration Group on page 199
- apply-groups on page 236
- apply-groups-except on page 237

Copyright © 2014, Juniper Networks, Inc.

help

Syntax help <(apropos *string* | reference <*statement-name*> | syslog <*syslog-tag*> | tip cli *number* | topic <*word*>)>

Release Information Command introduced before Junos OS Release 7.4.

Description Display help about available configuration statements or general information about getting help.

Options

apropos *string*—(Optional) Display statement names and help text that matches the string specified. If the string contains spaces, enclose it in quotation marks (" "). You can also specify a regular expression for the string, using standard UNIX-style regular expression syntax.

reference <*statement-name*>—(Optional) Display summary information for the statement. This information is based on summary descriptions that appear in the Junos configuration guides.

syslog <*syslog-tag*>—(Optional) Display information about system log messages.

tip cli *number*—(Optional) Display a tip about using the CLI. Specify the number of the tip you want to view.

topic <*word*>—(Optional) Display usage guidelines for a topic or configuration statement. This information is based on subjects that appear in the Junos configuration guides.

Entering the **help** command without an option provides introductory information about how to use the **help** command.

Required Privilege Level configure—To enter configuration mode.

Related Documentation

- Getting Online Help from the Junos OS Command-Line Interface on page 49

Copyright © 2014, Juniper Networks, Inc.

insert

Syntax insert <*statement-path*> *identifier1* (before | after) *identifier2*

Release Information Command introduced before Junos OS Release 7.4.

Description Insert an identifier in to an existing hierarchy.

Options

after—Place *identifier1* after *identifier2*.

before—Place *identifier1* before *identifier2*.

identifier1—Existing identifier.

identifier2—New identifier to insert.

statement-path—(Optional) Path to the existing identifier.

Required Privilege Level configure—To enter configuration mode, but other required privilege levels depend on where the statement is located in the configuration hierarchy.

Related Documentation

- Inserting a New Identifier in a Junos OS Configuration on page 75

Copyright © 2014, Juniper Networks, Inc.

load

Syntax load (factory-default | merge | override | patch | replace | set | update)
load (***filename*** | terminal) <relative>

QFX Series load (dhcp-snooping *filename*)

Release Information Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 11.1 for the QFX Series.

Description Load a configuration from an ASCII configuration file, from terminal input, or from the factory default. Your current location in the configuration hierarchy is ignored when the load operation occurs.

Options **dhcp-snooping**—(QFX Series switches) Loads DHCP snooping entries.

factory-default—Loads the factory configuration. The factory configuration contains the manufacturer's suggested configuration settings. The factory configuration is the router or switch's first configuration and is loaded when the router or switch is first installed and powered on.

NOTE: To load the factory default configuration, you must first unprotect any protected hierarchies in the configuration.

On J Series Services Routers, pressing and holding down the Config button on the router for 15 seconds causes the factory configuration to be loaded and committed. However, this operation deletes all other configurations on the router; using the **load factory-default** command does not.

filename—Name of the file to load. For information about specifying the filename, see "Specifying Filenames and URLs" on page 162.

merge—Combine the configuration that is currently shown in the CLI with the configuration.

override—Discard the entire configuration that is currently shown in the CLI and load the entire configuration. Marks every object as changed.

patch—Change part of the configuration and mark only those parts as changed.

replace—Look for a **replace** tag in ***filename***, delete the existing statement of the same name, and replace it with the configuration.

set—Merge a set of commands with an existing configuration. This option executes the configuration instructions line by line as they are stored in a file or from a terminal. The instructions can contain any configuration mode command, such as **set**, **edit**, **exit**, and **top**.

relative—(Optional) Use the **merge** or **replace** option without specifying the full hierarchy level.

Copyright © 2014, Juniper Networks, Inc.

terminal—Use the text you type at the terminal as input to the configuration. Type Ctrl+d to end terminal input.

update—Discard the entire configuration that is currently shown in the CLI, and load the entire configuration. Marks changed objects only.

NOTE: If you are using Junos OS in a Common Criteria environment, system log messages are created whenever a secret attribute is changed (for example, password changes or changes to the RADIUS shared secret). These changes are logged during the following configuration load operations:

```
load merge
load replace
load override
load update
```

For more information, see the *Secure Configuration Guide for Common Criteria and Junos-FIPS* .

Required Privilege Level configure—To enter configuration mode, but other required privilege levels depend on where the statement is located in the configuration hierarchy.

Related Documentation
- Loading a Configuration from a File on page 129

maximum-aggregate-pool (Batch Commits)

Syntax maximum-aggregate-pool *maximum-number-of-commits-to-aggregate*;

Hierarchy Level [edit system commit server],
[edit system commit synchronize server]

Release Information Statement introduced in Junos OS Release 12.1.

Description For Junos OS batch commits, specify the maximum number of individual commit operations that are aggregated or merged into a single commit operation.

Options *maximum-number-of-commits-to-aggregate*—Maximum number of individual commit operations that are aggregated or merged into a single commit operation.

Range: 1 through 4294967295

Default: 5

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring Batch Commit Server Properties on page 107

Copyright © 2014, Juniper Networks, Inc.

maximum-entries (Batch Commits)

Syntax maximum-entries *number-of-entries*;

Hierarchy Level [edit system commit server],
[edit system commit synchronize server]

Release Information Statement introduced in Junos OS Release 12.1.

Description For Junos OS batch commits, specify the maximum number of commit jobs that are included in the commit queue.

Options ***number-of-entries***—Maximum number of commit jobs that are included in the commit queue.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Batch Commit Server Properties on page 107

Copyright © 2014, Juniper Networks, Inc.

protect

Syntax protect (*hierarchy* | *statement* | *identifier*)

Release Information Command introduced in Junos OS Release 11.2.

Description Protect a hierarchy, statement, or identifier from modification or deletion.

Options none

Required Privilege Level configure—To enter configuration mode, but other required privilege levels depend on where the statement is located in the configuration hierarchy.

Related Documentation

- Example: Protecting the Junos OS Configuration from Modification or Deletion on page 137

Copyright © 2014, Juniper Networks, Inc.

quit

Syntax quit <configuration-mode>

Release Information Command introduced before Junos OS Release 7.4.

Description Exit the current level of the statement hierarchy, returning to the level prior to the last **edit** command, or exit from configuration mode. The **quit** and **exit** commands are synonyms.

Options none—Return to the previous edit level. If you are at the top of the statement hierarchy, exit configuration mode.

configuration-mode—(Optional) Exit from configuration mode.

Required Privilege Level configure—To enter configuration mode, but other required privilege levels depend on where the statement is located in the configuration hierarchy.

Related Documentation

- top on page 277
- up on page 280
- Displaying the Current Junos OS Configuration on page 87

Copyright © 2014, Juniper Networks, Inc.

rename

Syntax rename <*statement-path*> *identifier1* to *identifier2*

Release Information Command introduced before Junos OS Release 7.4.

Description Rename an existing configuration statement or identifier.

Options *identifier1*—Existing identifier to rename.

identifier2—New name of identifier.

statement-path—(Optional) Path to an existing statement or identifier.

NOTE: For example, to rename interface ge-0/0/0.0 to ge-0/0/10.0 at the following hierarchy level:

```
logical-systems {
  logical-system-abc {
  (...)
    protocols {
      ospf {
        area 0.0.0.0 {
        interface ge-0/1/0.0;
```

Issue the following command:

rename logical-systems *logical-system-abc* protocols ospf area *0.0.0.0* interface *ge-0/1/0.0.0* to *interface ge-0/1/10.0*

Required Privilege Level configure—To enter configuration mode, but other required privilege levels depend on where the statement is located in the configuration hierarchy.

Related Documentation

- Renaming an Identifier in a Junos OS Configuration on page 75

 Copyright © 2014, Juniper Networks, Inc.

replace

Syntax replace pattern ***pattern1*** with ***pattern2*** <upto ***n***>

Release Information Command introduced in Junos OS Release 7.6.

Description Replace identifiers or values in a configuration.

Options ***pattern1***—Text string or regular expression that defines the identifiers or values you want to match.

pattern2—Text string or regular expression that replaces the identifiers and values located with ***pattern1***.

Juniper Networks uses standard UNIX-style regular expression syntax (as defined in POSIX 1003.2). If the regular expression contains spaces, operators, or wildcard characters, enclose the expression in quotation marks. Greedy qualifiers (match as much as possible) are supported. Lazy qualifiers (match as little as possible) are not.

upto *n*—Number of objects replaced. The value of ***n*** controls the total number of objects that are replaced in the configuration (not the total number of times the pattern occurs). Objects at the same hierarchy level (siblings) are replaced first. Multiple occurrences of a pattern within a given object are considered a single replacement. If you do not specify an **upto** option, all identifiers and values in the configuration that match ***pattern1*** are replaced.

Required Privilege Level configure—To enter configuration mode, but other required privilege levels depend on where the statement is located in the configuration hierarchy.

Related Documentation

- Using Global Replace in a Junos OS Configuration on page 189

Copyright © 2014, Juniper Networks, Inc.

rollback

Syntax rollback <*number* | rescue>

Release Information Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 11.1 for the QFX Series.

Description Return to a previously committed configuration. The software saves the last 50 committed configurations, including the rollback number, date, time, and name of the user who issued the **commit** configuration command.

The currently operational Junos OS configuration is stored in the file **juniper.conf**, and the last three committed configurations are stored in the files **juniper.conf.1**, **juniper.conf.2**, and **juniper.conf.3**. These four files are located in the directory **/config**, which is on the router's flash drive. The remaining 46 previous committed configurations, the files **juniper.conf.4** through **juniper.conf.49**, are stored in the directory **/var/db/config**, which is on the router's hard disk.

During rollback, the configuration you specify is loaded from the associated file. Only objects in the rollback configuration that differ from the previously loaded configuration are marked as changed (equivalent to **load update**).

Options none (Optional)—Return to the most recently saved configuration.

number—(Optional) Configuration to return to. The range of values is from **0** through **49**. The most recently saved configuration is number 0, and the oldest saved configuration is number 49. The default is **0**.

rescue—(Optional) Return to the rescue configuration.

Required Privilege Level rollback—To roll back to configurations other than the one most recently committed.

Related Documentation

- Returning to a Previously Committed Junos OS Configuration on page 120
- Creating and Returning to a Rescue Configuration on page 123

Copyright © 2014, Juniper Networks, Inc.

run

Syntax run *command*

Release Information Command introduced before Junos OS Release 7.4.

Description Run a top-level CLI command without exiting from configuration mode.

Options *command*—CLI top-level command.

Required Privilege Level configure—To enter configuration mode.

Related Documentation
- Understanding Junos OS CLI Configuration Mode on page 58

Copyright © 2014, Juniper Networks, Inc.

save

Syntax save *filename*

QFX Series save (dhcp-snooping *filename*)

Release Information Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 11.1 for the QFX Series.

Description Save the configuration to an ASCII file. The contents of the current level of the statement hierarchy (and below) are saved, along with the statement hierarchy containing it. This allows a section of the configuration to be saved, while fully specifying the statement hierarchy.

When saving a file to a remote system, the software uses the **scp**/**ssh** protocol.

Options *filename*—Name of the saved file. You can specify a filename in one of the following ways:

- *filename*—File in the user's home directory (the current directory) on the local flash drive.
- *path/filename*—File on the local flash drive.
- /var/*filename* or /var/*path/filename*—File on the local hard disk.
- a:*filename* or a:*path/filename*—File on the local drive. The default path is / (the root-level directory). The removable media can be in MS-DOS or UNIX (UFS) format.
- *hostname:/path/filename, hostname:filename, hostname:path/filename,* or scp://*hostname/path/filename*—File on an **scp**/**ssh** client. This form is not available in the worldwide version of Junos OS. The default path is the user's home directory on the remote system. You can also specify *hostname* as *username*@*hostname*.
- ftp://*hostname*/*path*/*filename*—File on an FTP server. You can also specify *hostname* as *username* @*hostname* or *username*:*password* @*hostname*. The default path is the user's home directory. To specify an absolute path, the path must start with the string **%2F**; for example, ftp://*hostname*/%2F*path/filename*. To have the system prompt you for the password, specify **prompt** in place of the password. If a password is required, and you do not specify the password or **prompt**, an error message is displayed:

  user@host> **file copy ftp://*username*@ftp.*hostname*.net//*filename***
  file copy ftp.*hostname*.net: Not logged in.
  user@host> **file copy ftp://*username*:prompt@ftp*hostname*.net//*filename***

  Password for *username*@ftp.*hostname*.net:

- http://*hostname*/*path*/*filename*—File on a Hypertext Transfer Protocol (HTTP) server. You can also specify *hostname* as *username*@*hostname* or *username*:*password*@*hostname*. If a password is required and you omit it, you are prompted for it.
- re0:*/path/filename* or re1:*/path/filename*—File on a local Routing Engine.

Copyright © 2014, Juniper Networks, Inc.

Required Privilege Level configure—To enter configuration mode.

Related Documentation
- Deactivating and Reactivating Statements and Identifiers in a Junos OS Configuration on page 82

server (Batch Commits)

Syntax

```
server {
  commit-interval<number-of-seconds-between-commits>;
  days-to-keep-error-logs<days-to-keep-error-log-entries>;
  maximum-aggregate-pool<maximum-number-of-commits-to-aggregate>;
  maximum-entries <number-of-entries>;
  traceoptions {
    file filename;
    files number;
    flag (all | batch | commit-server | configuration);
    size maximum-file-size;
    (world-readable | no-world-readable);
  }
}
```

Hierarchy Level [edit system commit]

Release Information Statement introduced in Junos OS Release 12.1.

Description Configure the system commit to occur in batches. Configure parameters for aggregating and saving batch commits.

Options

commit-interval—Configure the interval between commits.

days-to-keep-error-logs—Configure the number of days to keep log entries.

maximum-aggregate-pool—Configure the maximum number of commits to aggregate together.

maximum-entries —Configure the maximum number of commit entries.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring Batch Commit Server Properties on page 107

Copyright © 2014, Juniper Networks, Inc.

set

Syntax set <*statement-path*> *identifier*

Release Information Command introduced before Junos OS Release 7.4.

Description Create a statement hierarchy and set identifier values. This is similar to **edit** except that your current level in the hierarchy does not change.

Options *identifier*—Name of the statement or identifier to set.

statement-path—(Optional) Path to an existing statement hierarchy level. If that hierarchy level does not exist, it is created.

Required Privilege Level configure—To enter configuration mode, but other required privilege levels depend on where the statement is located in the configuration hierarchy.

Related Documentation

- edit on page 249
- Displaying the Current Junos OS Configuration on page 87

Copyright © 2014, Juniper Networks, Inc.

show

Syntax show <*statement-path*> <*identifier*>

Release Information Command introduced before Junos OS Release 7.4.

Description Display the current configuration.

Options none—Display the entire configuration at the current hierarchy level.

identifier—(Optional) Display the configuration for the specified identifier.

statement-path—(Optional) Display the configuration for the specified statement hierarchy path.

Required Privilege Level configure—To enter configuration mode, but other required privilege levels depend on where the statement is located in the configuration hierarchy.

Related Documentation

- show | display inheritance on page 271
- show | display omit on page 272
- *show | display rfc5952*
- *show | display rfc5952*
- show | display set on page 273
- show | display set relative on page 274
- show groups junos-defaults on page 275
- Displaying the Current Junos OS Configuration on page 87

Copyright © 2014, Juniper Networks, Inc.

show configuration

Syntax show configuration
<*statement-path*>

Release Information Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.

Description Display the configuration that currently is running on the router or switch, which is the last committed configuration.

Options **none**—Display the entire configuration.

statement-path—(Optional) Display one of the following hierarchies in a configuration. (Each ***statement-path*** option has additional suboptions not described here. See the appropriate configuration guide or EX Series switch documentation for more information.)

- **access**—Network access configuration.
- **access-profile**—Access profile configuration.
- **accounting-options**—Accounting data configuration.
- **applications**—Applications defined by protocol characteristics.
- **apply-groups**—Groups from which configuration data is inherited.
- **chassis**—Chassis configuration.
- **chassis network-services**—Current running mode.
- **class-of-service**—Class-of-service configuration.
- **diameter**—Diameter base protocol layer configuration.
- **ethernet-switching-options**—(EX Series switch only) Ethernet switching configuration.
- **event-options**—Event processing configuration.
- **firewall**—Firewall configuration.
- **forwarding-options**—Options that control packet sampling.
- **groups**—Configuration groups.
- **interfaces**—Interface configuration.
- **jsrc**—JSRC partition configuration.
- **jsrc-partition**—JSRC partition configuration.
- **logical-systems**—Logical system configuration.
- **poe**—(EX Series switch only) Power over Ethernet configuration.
- **policy-options**—Routing policy option configuration.
- **protocols**—Routing protocol configuration.

Copyright © 2014, Juniper Networks, Inc.

- **routing-instances**—Routing instance configuration.
- **routing-options**—Protocol-independent routing option configuration.
- **security**—Security configuration.
- **services**—Service PIC applications configuration.
- **snmp**—Simple Network Management Protocol configuration.
- **system**—System parameters configuration.
- **virtual-chassis**—(EX Series switch only) Virtual Chassis configuration.
- **vlans**—(EX Series switch only) VLAN configuration.

Additional Information

The portions of the configuration that you can view depend on the user class that you belong to and the corresponding permissions. If you do not have permission to view a portion of the configuration, the text **ACCESS-DENIED** is substituted for that portion of the configuration. If you do not have permission to view authentication keys and passwords in the configuration, because the secret permission bit is not set for your user account, the text SECRET-DATA is substituted for that portion of the configuration. If an identifier in the configuration contains a space, the identifier is displayed in quotation marks.

Likewise, when you issue the **show configuration** command with the **| display set** pipe option to view the configuration as **set** commands, those portions of the configuration that you do not have permissions to view are substituted with the text **ACCESS-DENIED**.

Required Privilege Level

view

Related Documentation

- Displaying the Current Junos OS Configuration on page 87
- Overview of Junos OS CLI Operational Mode Commands on page 149

List of Sample Output

show configuration on page 269
show configuration policy-options on page 270

Output Fields

This command displays information about the current running configuration.

Sample Output

show configuration

```
user@host> show configuration
## Last commit: 2006-10-31 14:13:00 PST by alant version "8.2I0 [builder]"; ##
last changed: 2006-10-31 14:05:53 PST
system {
    host-name nestor;
    domain-name east.net;
    backup-router 192.1.1.254;
   time-zone America/Los_Angeles;
    default-address-selection;
    name-server {
        192.154.169.254;
        192.154.169.249;
```

Copyright © 2014, Juniper Networks, Inc.

```
            192.154.169.176;
        }
        services {
            telnet;
        }
        tacplus-server {
            1.2.3.4 {
                secret /* SECRET-DATA */;
                ...
            }
        }
    }
    interfaces {
        ...
    }
    protocols {
            isis {
                export "direct routes";
            }
    }
    policy-options {
        policy-statement "direct routes" {
            from protocol direct;
            then accept;
        }
    }
```

show configuration policy-options

```
user@host> show configuration policy-options
policy-options {
    policy-statement "direct routes" {
        from protocol direct;
        then accept;
    }
}
```


Copyright © 2014, Juniper Networks, Inc.

show | display inheritance

Syntax show | display inheritance <brief | defaults | no-comments | terse>

Release Information Command introduced before Junos OS Release 7.4.

Description Show the inherited configuration data and information about the source group from which the configuration has been inherited. Show interface ranges configuration data in expanded format and information about the source interface-range from which the configuration has been expanded

```
user@host# show system ports | display inheritance defaults
## 'console' was inherited from group 'junos-defaults'
## 'vt100' was inherited from group 'junos-defaults'
## console type vt100;
```

```
user@host# show system login class readonly | display inheritance
## 'interface' was inherited from group global'
## 'network' was inherited from group global'
## 'routing' was inherited from group global'
## 'system' was inherited from group global'
## 'trace' was inherited from group global'
## 'view' was inherited from group global'
##
permissions [ interface network routing system trace view ];
```

```
user@host# show system login class readonly | display inheritance no-comments
permissions [ interface network routing system trace view ];
```

Options

- **brief**—Display brief output for the command.
- **defaults**—Display the Junos OS defaults that have been applied to the configuration.
- **no-comments**—Display configuration information without inline comments marked with **##**.
- **terse**—Display terse output with inheritance details as inline comment.

Required Privilege Level view

Related Documentation

- Using Junos OS Defaults Groups on page 224

Copyright © 2014, Juniper Networks, Inc.

show | display omit

Syntax show | display omit

Release Information Command introduced in Junos OS Release 8.2.

Description Display configuration statements (including those marked as hidden by the **apply-flags omit** configuration statement).

```
user@host# show | display omit
  system {
    apply-flags omit;
    login {
      message lengthy-login-message;
    }
  }
```

Required Privilege Level view

Related Documentation

- show on page 267

 Copyright © 2014, Juniper Networks, Inc.

show | display set

Syntax show | display set

Release Information Command introduced before Junos OS Release 7.4.

Description Display the configuration as a series of configuration mode commands required to re-create the configuration from the top level of the hierarchy as **set** commands

```
user@host# show | display set
   set interfaces fe-0/0/0 unit 0 family inet address 192.168.1.230/24
   set interfaces fe-0/0/0 unit 0 family iso
   set interfaces fe-0/0/0 unit 0 family mpls
   set interfaces fe-0/0/0 unit 1 family inet address 10.0.0.1/8
deactivate interfaces fe-0/0/0 unit 1
```

Required Privilege Level view

Related Documentation

- show on page 267
- Displaying set Commands from the Junos OS Configuration on page 92

Copyright © 2014, Juniper Networks, Inc.

show | display set relative

Syntax show | display set relative

Release Information Command introduced before Junos OS Release 7.4.

Description Display the configuration as a series of configuration mode commands required to re-create the configuration from the current hierarchy level.

```
[edit interfaces fe-0/0/0]
  user@host# show
  unit 0 {
    family inet {
      address 192.107.1.230/24;
    }
    family iso;
    family mpls;
  }
  inactive: unit 1 {
    family inet {
      address 10.0.0.1/8;
    }
  }
  user@host# show | display set relative
  set unit 0 family inet address 192.107.1.230/24
  set unit 0 family iso
  set unit 0 family mpls
  set unit 1 family inet address 10.0.0.1/8
deactivate unit 1
```

Required Privilege Level view

Related Documentation

- Displaying set Commands from the Junos OS Configuration on page 92

Copyright © 2014, Juniper Networks, Inc.

show groups junos-defaults

Syntax show groups junos-defaults

Release Information Command introduced before Junos OS Release 7.4.

Description Display the full set of available preset statements from the Junos OS defaults group.

```
user@host# show groups junos-defaults
  groups {
    junos-defaults {
      applications {
        # File Transfer Protocol
        application junos-ftp {
          application-protocol ftp;
          protocol tcp;
          destination-port 21;
        }
        # Trivial File Transfer Protocol
        application junos-tftp {
          application-protocol tftp;
          protocol udp;
          destination-port 69;
        }
        # RPC port mapper on TCP
        application junos-rpc-portmap-tcp {
          application-protocol rpc-portmap;
          protocol tcp;
          destination-port 111;
        }
        # RPC port mapper on UDP
      }
    }
  }
```

Required Privilege Level view

Related Documentation

- Using Junos OS Defaults Groups on page 224

Copyright © 2014, Juniper Networks, Inc.

status

Syntax status

Release Information Command introduced before Junos OS Release 7.4.

Description Display the users currently editing the configuration.

Required Privilege Level configure—To enter configuration mode.

- "Displaying Users Currently Editing the Junos OS Configuration" on page 94.

Copyright © 2014, Juniper Networks, Inc.

top

Syntax top <*configuration-command*>

Release Information Command introduced before Junos OS Release 7.4.

Description Return to the top level of configuration command mode, which is indicated by the **[edit]** banner.

Options ***configuration-command***—(Optional) Issue configuration mode commands from the top of the hierarchy.

Required Privilege Level configure—To enter configuration mode.

Related Documentation

- Displaying the Current Junos OS Configuration on page 87
- exit on page 250
- up on page 280

Copyright © 2014, Juniper Networks, Inc.

traceoptions (Batch Commits)

Syntax

```
traceoptions {
  file filename;
  files number;
  flag (all | batch | commit-server | configuration);
  size maximum-file-size;
  (world-readable | no-world-readable);
}
```

Hierarchy Level

[edit system commit server],
[edit system commit synchronize server]

Release Information

Statement introduced in Junos OS Release 12.1.

Description

For Junos OS batch commits, configure tracing operations.

Options

file *name*—Name of the file to receive the output of the tracing operation.

NOTE: If you configure traceoptions and do not explicitly specify a filename for logging the events, the batch commit events are logged in the commitd file (var/log/commitd) by default.

files *number*—Maximum number of trace files.

flag *flag*—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements. You can include the following flags:

- **all**—All tracing operations flags.
- **batch**—Tracing operations for batch events.
- **commit-server**—Tracing operations for commit server events.
- **configuration**—Tracing operations for the reading of configuration.

size—Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB).

world-readable | no-world-readable—readable—Grant all users permission to read archived log files, or restrict the permission only to the root user and users who have the Junos OS maintenance permission.

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Batch Commit Server Properties on page 107

Copyright © 2014, Juniper Networks, Inc.

unprotect

Syntax unprotect (*hierarchy* | *statement* | *identifier*)

Release Information Command introduced in Junos OS Release 11.2.

Description Unprotect a protected hierarchy, configuration statement, or an identifier.

Options none

Required Privilege Level configure—To enter configuration mode, but other required privilege levels depend on where the statement is located in the configuration hierarchy.

Related Documentation

- top on page 277
- up on page 280
- Displaying the Current Junos OS Configuration on page 87

Copyright © 2014, Juniper Networks, Inc.

up

Syntax up <*number*> <*configuration-command*>

Release Information Command introduced before Junos OS Release 7.4.

Description Move up one level in the statement hierarchy.

Options none—Move up one level in the configuration hierarchy.

configuration-command—(Optional) Issue configuration mode commands from a location higher in the hierarchy.

number—(Optional) Move up the specified number of levels in the configuration hierarchy.

Required Privilege Level configure—To enter configuration mode.

Related Documentation

- Displaying the Current Junos OS Configuration on page 87
- exit on page 250
- top on page 277

Copyright © 2014, Juniper Networks, Inc.

update

Syntax update

Release Information Command introduced in Junos OS Release 7.5.

Description Update private candidate configuration with a copy of the most recently committed configuration, including your private changes.

NOTE: The update command is available only when you are in configure private mode.

Required Privilege Level configure—To enter configuration mode.

Related Documentation

- Updating the configure private Configuration on page 86.

Copyright © 2014, Juniper Networks, Inc.

when

Syntax

```
when {
  chassis chassis-id;
  member member-id;
  model model-id;
  node node-id;
  routing-engine routing-engine-id;
  time <start-time> [to <end-time>];
}
```

Hierarchy Level [edit groups *group-name*]

Release Information Statement introduced in Junos OS Release 11.3.

Description Define conditions under which the configuration group should be applied. Conditions include the type of chassis, model, or Routing Engine, virtual chassis member, cluster node, and start and optional end time of day. If you specify multiple conditions in a single configuration group, all conditions must be met before the configuration group is applied.

Options

chassis *chassis-id*—Specify the chassis type of the router. Valid types include SCC0, SCC1, LCC0, LCC1 ... LCC3.

member *member-id*—Specify the name of the member of the virtual chassis.

model *model-id*—Specify the model name of the router, such as m7i or tx100.

node *node-id*—Specify the cluster node.

routing-engine *routing-engine-id*—Specify the type of Routing Engine, re0 or re1.

time <*start-time*> [to <*end-time*>]—Specify the start time or time duration for this configuration group to be applied. If only the start time is specified, the configuration group is applied at the specified time and remains in effect until the time is changed. If the end time is specified, then on each day, the applied configuration group is started and stopped at the specified times. The syntax for specifying the time is: **time <*start-time*> [to <*end-time*>]** using the time format yyyy-mm-dd.hh:mm, hh:mm, or hh.

Required Privilege Level configure—To enter configuration mode.

Related Documentation

- Creating a Junos OS Configuration Group on page 199
- apply-groups on page 236
- apply-groups-except on page 237
- groups on page 251

Copyright © 2014, Juniper Networks, Inc.

wildcard delete

Syntax wildcard delete <*statement-path*> <*identifier*> <*regular-expression*>

Release Information Command introduced before Junos OS Release 7.4.

Description Delete a statement or identifier. All subordinate statements and identifiers contained within the specified statement path are deleted with it.

Deleting a statement or an identifier effectively "unconfigures" or disables the functionality associated with that statement or identifier.

If you do not specify ***statement-path*** or ***identifier***, the entire hierarchy starting at the current hierarchy level is removed.

Options ***identifier***—(Optional) Name of the statement or identifier to delete.

regular-expression—(Optional) The pattern based on which you want to delete multiple items. When you use the **wildcard** command to delete related configuration items, the ***regular-expression*** must be the final statement.

statement-path—(Optional) Path to an existing statement or identifier. Include this if the statement or identifier to be deleted is not at the current hierarchy level.

Required Privilege Level configure—To enter configuration mode. Other required privilege levels depend on where the statement is located in the configuration hierarchy.

Related Documentation

- Example: Using Global Replace in a Junos OS Configuration—Using the upto Option on page 192.

Copyright © 2014, Juniper Networks, Inc.

Copyright © 2014, Juniper Networks, Inc.

CHAPTER 13

Junos OS CLI Environment Commands

- set cli complete-on-space
- set cli directory
- set cli idle-timeout
- set cli prompt
- set cli restart-on-upgrade
- set cli screen-length
- set cli screen-width
- set cli terminal
- set cli timestamp
- set date
- show cli
- show cli
- show cli authorization
- show cli directory
- show cli history

Copyright © 2014, Juniper Networks, Inc.

set cli complete-on-space

Syntax set cli complete-on-space (off | on)

Release Information Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.

Description Set the command-line interface (CLI) to complete a partial command entry when you type a space or a tab. This is the default behavior of the CLI.

Options **off**—Turn off command completion.

on—Allow either a space or a tab to be used for command completion.

Required Privilege Level view

Related Documentation
- *CLI User Interface Overview*
- show cli on page 296

List of Sample Output set cli complete-on-space on page 286

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

set cli complete-on-space

In the following example, pressing the Spacebar changes the partial command entry from **com** to **complete-on-space**. The example shows how adding the keyword **off** at the end of the command disables command completion.

```
user@host> set cli com<Space>
user@host>set cli complete-on-space off
Disabling complete-on-space
```


Copyright © 2014, Juniper Networks, Inc.

set cli directory

Syntax set cli directory *directory*

Release Information Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.

Description Set the current working directory.

Options ***directory***—Pathname of the working directory.

Required Privilege Level view

Related Documentation
- *CLI User Interface Overview*
- show cli directory on page 301

List of Sample Output set cli directory on page 287

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

set cli directory

```
user@host> set cli directory /var/home/regress
Current directory: /var/home/regress
```

set cli idle-timeout

Syntax set cli idle-timeout
<*minutes*>

Release Information Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.

Description Set the maximum time that an individual session can be idle before the user is logged off the router or switch.

Options ***minutes***—(Optional) Maximum idle time. The range of values, in minutes, is 0 through 100,000. If you do not issue this command, and the user's login class does not specify this value, the user is never forced off the system after extended idle times. Setting the value to 0 disables the timeout.

Required Privilege Level view

Related Documentation
- *CLI User Interface Overview*
- show cli on page 296

List of Sample Output set cli idle-timeout on page 288

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

set cli idle-timeout

```
user@host> set cli idle-timeout 60
Idle timeout set to 60 minutes
```


Copyright © 2014, Juniper Networks, Inc.

set cli prompt

Syntax set cli prompt *string*

Release Information Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.

Description Set the prompt so that it is displayed within the CLI.

Options ***string***—CLI prompt string. To include spaces in the prompt, enclose the string in quotation marks. By default, the string is ***username*@*hostname***.

Required Privilege Level view

Related Documentation
- *CLI User Interface Overview*
- show cli on page 296

List of Sample Output set cli prompt on page 289

Output Fields When you enter this command, the new CLI prompt is displayed.

Sample Output

set cli prompt

```
user@host> set cli prompt lab1-router>
lab1-router>
```

Copyright © 2014, Juniper Networks, Inc.

set cli restart-on-upgrade

Syntax set cli restart-on-upgrade string (off | on)

Release Information Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.

Description For an individual session, set the CLI to prompt you to restart the router or switch after upgrading the software.

Options **off**—Disables the prompt.

on—Enables the prompt.

Required Privilege Level view

Related Documentation
- *CLI User Interface Overview*
- show cli on page 296

List of Sample Output set cli restart-on-upgrade on page 290

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

set cli restart-on-upgrade

```
user@host> set cli restart-on-upgrade on
Enabling restart-on-upgrade
```


Copyright © 2014, Juniper Networks, Inc.

set cli screen-length

Syntax set cli screen-length *length*

Release Information Command introduced before Junos OS Release 7.4.

Description Set terminal screen length.

```
user@host> set cli screen-length 75
Screen length set to 75
```

Options ***length***—Number of lines of text that the terminal screen displays. The range of values, in number of lines, is **24** through **100,000**. The default is **24**.

The point at which the **---(more)---** prompt appears on the screen is a function of this setting and the settings for the **set cli screen-width** and **set cli terminal** commands.

Required Privilege Level view

Related Documentation

- Setting the Screen Length on page 231
- Understanding the Screen Length and Width Settings on page 232
- set cli screen-width on page 292
- *set cli terminal*
- show cli on page 298

Copyright © 2014, Juniper Networks, Inc.

set cli screen-width

Syntax set cli screen-width ***width***

Release Information Command introduced before Junos OS Release 7.4.

Description Set the terminal screen width.

```
user@host> set cli screen-width
Screen width set to 132
```

Options ***width***—Number of characters in a line. The range of values is **40** through **1024**. The default is **80**.

The point at which the **---(more)---** prompt appears on the screen is a function of this setting and the settings for the **set cli screen-length** and **set cli terminal** commands.

NOTE: In Junos OS Release 13.3 and later, the range of *width* is 40 through 1024. In earlier Junos releases, the range is 0 through 1024.

Required Privilege Level view

Related Documentation

- Setting the Screen Width on page 232
- set cli screen-length on page 291
- *set cli terminal*
- show cli on page 298

Copyright © 2014, Juniper Networks, Inc.

set cli terminal

Syntax set cli terminal *terminal-type*

Release Information Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.

Description Set the terminal type.

Options ***terminal-type***—Type of terminal that is connected to the Ethernet management port:

- **ansi**—ANSI-compatible terminal (80 characters by 24 lines)
- **small-xterm**—Small xterm window (80 characters by 24 lines)
- **vt100**—VT100-compatible terminal (80 characters by 24 lines)
- **xterm**—Large xterm window (80 characters by 65 lines)

Required Privilege Level view

Related Documentation

- *CLI User Interface Overview*
- *set cli screen-length*
- *set cli screen-width*
- show cli on page 296

List of Sample Output set cli terminal on page 293

Output Fields This command provides no output.

Sample Output

set cli terminal

```
user@host> set cli terminal xterm
```

set cli timestamp

Syntax set cli timestamp (format *timestamp-format* | disable)

Release Information Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.

Description Set a timestamp for CLI output.

Options **format *timestamp-format***—Set the date and time format for the timestamp. The timestamp format you specify can include the following placeholders in any order:

- **%m**—Two-digit month
- **%d**—Two-digit date
- **%T**—Six-digit hour, minute, and seconds

disable—Remove the timestamp from the CLI.

NOTE: A timestamp is displayed by default when no command output is generated.

Required Privilege Level view

Related Documentation

- *CLI User Interface Overview*
- show cli on page 296

List of Sample Output set cli timestamp on page 294

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

set cli timestamp

```
user@host> set cli timestamp format '%m-%d-%T'
'04-21-17:39:13'
CLI timestamp set to: '%m-%d-%T'
```


Copyright © 2014, Juniper Networks, Inc.

set date

Syntax set date (***date-time*** | ntp <***ntp-server***> <source-address ***source-address***>)

Release Information Command introduced before Junos OS Release 7.4.

Description Set the date and time.

```
user@host> set date ntp
21 Apr 17:22:02 ntpdate[3867]: step time server 172.17.27.46 offset 8.759252 sec
```

Options

- ***date-time***—Specify date and time in one of the following formats:
  - ***YYYYMMDDHHMM.SS***
  - **"*month DD*, *YYYY HH:MM*(am | pm)"**
- **ntp**—Configure the router to synchronize the current date and time setting with a Network Time Protocol (NTP) server.
- ***ntp-server*—(**Optional) Specify the IP address of one or more NTP servers.
- **source-address *source-address***—(Optional) Specify the source address that is used by the router to contact the remote NTP server.

Required Privilege Level view

Related Documentation

- *Setting the Date and Time Locally*

Copyright © 2014, Juniper Networks, Inc.

show cli

List of Syntax
Syntax on page 296
Syntax (QFX Series) on page 296

Syntax
show cli

Syntax (QFX Series)
show cli
<authorization>
<directory>
<history *count*>

Release Information
Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.1 for the QFX Series.

Description
Display configured CLI settings.

Options
This command has no options.

Required Privilege Level
view

List of Sample Output
show cli on page 297

Output Fields
Table 16 on page 296 lists the output fields for the **show cli** command. Output fields are listed in the approximate order in which they appear.

Table 16: show cli Output Fields

Field Name	Field Description
CLI complete-on-space	Capability to complete a partial command entry when you type a space or a tab: **on** or **off**.
CLI idle-timeout	Maximum time that an individual session can be idle before the user is logged out from the router or switch. When this feature is enabled, the number of minutes is displayed. Otherwise, the state is **disabled**.
CLI restart-on-upgrade	CLI is set to prompt you to restart the router or switch after upgrading the software: **on** or **off**.
CLI screen-length	Number of lines of text that the terminal screen displays.
CLI screen-width	Number of characters in a line on the terminal screen.
CLI terminal	Terminal type.
CLI is operating in	Mode: **enhanced**.
CLI timestamp	Date and time format for the timestamp. If the timestamp is not set, the state is **disabled**.
CLI working directory	Pathname of the working directory.

Copyright © 2014, Juniper Networks, Inc.

Sample Output

show cli

```
user@host> show cli
CLI complete-on-space set to on
CLI idle-timeout disabled
CLI restart-on-upgrade set to on
CLI screen-length set to 47
CLI screen-width set to 132
CLI terminal is 'vt100'
CLI is operating in enhanced mode
CLI timestamp disabled
CLI working directory is '/var/home/regress'
```

Copyright © 2014, Juniper Networks, Inc.

show cli

Syntax show cli

Release Information Command introduced before Junos OS Release 7.4.

Description Display configured CLI settings.

```
user@host> show cli
CLI complete-on-space set to on
CLI idle-timeout disabled
CLI restart-on-upgrade set to on
CLI screen-length set to 47
CLI screen-width set to 132
CLI terminal is 'vt100'
CLI is operating in enhanced mode
CLI timestamp disabled
CLI working directory is '/var/home/regress'
```

Required Privilege Level view

Related Documentation

- show cli authorization on page 299
- *show cli directory*

Copyright © 2014, Juniper Networks, Inc.

show cli authorization

Syntax show cli authorization

Release Information Command introduced before Junos OS Release 7.4.

Description Display the permissions for the current user.

```
user@host> show cli authorization
Current user: 'root' login: ‘boojum’ class '(root)'
Permissions:
Permissions:
    admin       -- Can view user accounts
    admin-control-- Can modify user accounts
    clear       -- Can clear learned network info
    configure   -- Can enter configuration mode
    control     -- Can modify any config
    edit        -- Can edit full files
    field       -- Can use field debug commands
    floppy      -- Can read and write the floppy
    interface   -- Can view interface configuration
    interface-control-- Can modify interface configuration
    network     -- Can access the network
    reset       -- Can reset/restart interfaces and daemons
    routing     -- Can view routing configuration
    routing-control-- Can modify routing configuration
    shell       -- Can start a local shell
    snmp        -- Can view SNMP configuration
    snmp-control-- Can modify SNMP configuration
    system      -- Can view system configuration
    system-control-- Can modify system configuration
    trace       -- Can view trace file settings
    trace-control-- Can modify trace file settings
    view        -- Can view current values and statistics
    maintenance -- Can become the super-user
    firewall    -- Can view firewall configuration
    firewall-control-- Can modify firewall configuration
    secret      -- Can view secret statements
    secret-control-- Can modify secret statements
    rollback    -- Can rollback to previous configurations
    security    -- Can view security configuration
    security-control-- Can modify security configuration
    access      -- Can view access configuration
    access-control-- Can modify access configuration
    view-configuration-- Can view all configuration (not including secrets)
    flow-tap    -- Can view flow-tap configuration
    flow-tap-control-- Can modify flow-tap configuration
    idp-profiler-operation-- Can Profiler data
    pgcp-session-mirroring-- Can view pgcp session mirroring configuration
    pgcp-session-mirroring-control-- Can modify pgcp session mirroring
configuration
    storage     -- Can view fibre channel storage protocol configuration
    storage-control-- Can modify fibre channel storage protocol configuration
    all-control -- Can modify any configuration
```

Required Privilege Level view

Copyright © 2014, Juniper Networks, Inc.

Related Documentation

- show cli on page 298
- *show cli directory*

Copyright © 2014, Juniper Networks, Inc.

show cli directory

Syntax show cli directory

Release Information Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.1 for the QFX Series.

Description Display the current working directory.

Options This command has no options.

Required Privilege Level view

List of Sample Output show cli directory on page 301

Output Fields Table 17 on page 301 lists the output fields for the **show cli directory** command. Output fields are listed in the approximate order in which they appear.

Table 17: show cli directory Output Fields

Field Name	Field Description
Current directory	Pathname of the current working directory.

Sample Output

show cli directory

```
user@host> show cli directory
Current directory: /var/home/regress
```

show cli history

Syntax show cli history <*count*>

Release Information Command introduced before Junos OS Release 7.4.

Description Display a list of previous CLI commands.

```
user@host> show cli history
  11:14:14 -- show arp
  11:22:10 -- show cli authorization
  11:27:12 -- show cli history
```

Options none—Display all previous CLI commands.

count—(Optional) Maximum number of commands to display.

Required Privilege Level view

Related Documentation

- Displaying the Junos OS CLI Command and Word History on page 56

 Copyright © 2014, Juniper Networks, Inc.

CHAPTER 14

Junos OS CLI Operational Mode Commands

- configure
- file
- help
- | (pipe)
- request
- request system commit server pause
- request system commit server queue cleanup
- request system commit server start
- restart
- set
- show system commit server queue
- show system commit server status

Copyright © 2014, Juniper Networks, Inc.

configure

Syntax

configure
<batch>
<dynamic>
<exclusive>
<private>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.

Description

Enter configuration mode. When this command is entered without any optional keywords, everyone can make configuration changes and commit all changes made to the configuration.

Options

none—Enter configuration mode.

batch—(Optional) Work in the batch commit mode where commit operations are executed in batches.

dynamic—(Optional) Configure routing policies and certain routing policy objects in a dynamic database that is not subject to the same verification required in the standard configuration database. As a result, the time it takes to commit changes to the dynamic database is much shorter than for the standard configuration database. You can then reference these policies and policy objects in routing policies you configure in the standard database.

exclusive—(Optional) Lock the candidate configuration for as long as you remain in configuration mode, allowing you to make changes without interference from other users. Other users can enter and exit configuration mode, but they cannot change the configuration.

private—(Optional) Allow multiple users to edit different parts of the configuration at the same time and to commit only their own changes, or to roll back without interfering with one another's changes. You cannot commit changes in configure private mode when another user is in configure exclusive mode.

Additional Information

For more information about the different methods of entering configuration mode and the restrictions that apply, see the *Junos OS Administration Library for Routing Devices*.

Required Privilege Level

configure

Related Documentation

- show configuration on page 268

List of Sample Output

configure on page 305

Output Fields

When you enter this command, you are placed in configuration mode and the system prompt changes from ***hostname>*** to ***hostname#***.

Copyright © 2014, Juniper Networks, Inc.

Sample Output

configure

```
user@host> configure
Entering configuration mode
[edit]
user@host#
```

Copyright © 2014, Juniper Networks, Inc.

file

Syntax file <archive |change-owner | change-permission | checksum |compare | compress | copy | delete | delete-directory | link | list | make-directory | rename | show | source address>

Release Information Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 11.1 for the QFX Series.
change-owner, **change-permission**, **compress**, **delete-directory**, **link**, and **make-directory** options added in Junos OS Release 14.1.

Description Archive files from the device, copy files to and from the router or switch, calculate the file checksum, compare files, delete a file from the device, list files on the device, rename a file, show file contents, show the local address to initiate a connection, change owner of a file, change permission of a file, compress a file, delete a directory, create a link between files, or create a new directory.

Options

archive (Optional) —Archive, and optionally compress, one or multiple local system files as a single file, locally or at a remote location.

change-owner (Optional) —Change owner of a file.

change-permission (Optional) —Change permission of a file.

checksum (Optional) —Calculate the Message Digest 5 (MD5) checksum of a file.

compare (Optional) —Compare two local files and describe the differences between them in default, context, or unified output styles.

compress (Optional) —Compress a file.

copy (Optional) —Copy files from one place to another on the local switch or between the local switch and a remote system.

delete (Optional) —Delete a file on the local switch.

delete-directory (Optional) —Delete a directory.

link (Optional) —Create a link between files.

list (Optional) —Display a list of files on the local switch.

make-directory (Optional) —Create a new directory.

rename (Optional) —Rename a file on the local switch.

show (Optional) —Display the contents of a file.

source address (Optional) —Specify the source address of the local file.

Required Privilege Level maintenance

Copyright © 2014, Juniper Networks, Inc.

Related Documentation

- Viewing Files and Directories on a Device Running Junos OS on page 159
- CLI Explorer

Copyright © 2014, Juniper Networks, Inc.

help

Syntax help < (apropos *string* | reference <*statement-name*> | syslog <*syslog-tag*> | tip cli *number* | topic <*word*>)>

Release Information Command introduced before Junos OS Release 7.4.
apropos option added in Junos OS Release 8.0.

Description Display help about available operational commands, configuration statements, or general information about getting help. Entering the **help** command without an option provides introductory information about how to use the **help** and **?** commands.

Options

apropos *string*—(Optional) Display command names and help text that matches the string specified. If the string contains spaces, enclose it in quotation marks (" "). You can also specify a regular expression for the string, using standard UNIX-style regular expression syntax.

reference <*statement-name*>—(Optional) Display summary information for a configuration statement. This information is based on summary descriptions that appear in the Junos configuration guides.

syslog <*syslog-tag*>—(Optional) Display information about system log messages.

tip cli *number*—(Optional) Display a tip about using the CLI. Specify the number of the tip you want to view.

topic <*word*>—(Optional) Display usage guidelines for a topic or configuration statement. This information is based on subjects that appear in the Junos configuration guides.

Required Privilege Level None

Related Documentation

- Getting Online Help from the Junos OS Command-Line Interface on page 49

 Copyright © 2014, Juniper Networks, Inc.

| (pipe)

Syntax | (compare | count | display (changed | commit-scripts | detail | display set | inheritance | omit | xml) | except *pattern* | find *pattern* | hold | last *lines* | match *pattern*| no-more | refresh *interval* | request message (all | *account@terminal*) resolve <full-names> | save *filename*| append *filename* | | tee | trim *columns*)

Release Information Command introduced before Junos OS Release 7.4.
display commit-scripts option added in Junos OS Release 7.4.
tee option added in Junos OS Release 14.1.

Description Filter the output of an operational mode or a configuration mode command.

Options **append *filename***—Append the output to a file.

compare (filename | rollback *n*)—(Configuration mode only, and only with the **show** command) Compare configuration changes with another configuration file.

count—Display the number of lines in the output.

display—Display additional information about the configuration contents.

- **changed**—Tag changes with **junos:changed attribute** (XML only).
- **commit-scripts**—(Configuration mode only) Display all statements that are in a configuration, including statements that were generated by transient changes. For more information, see the *Junos OS Automation Library*.
- **detail**—(Configuration mode only) Display configuration data detail.
- **inheritance <brief | default | no-comments |groups | terse>**—(Configuration mode only) Display inherited configuration data and source group.
- **omit**—(Configuration mode only) Display configuration statements omitted by the **apply-flags omit** configuration statement.
- **set**—Display the configuration as a series of configuration mode commands required to re-create the configuration.
- **xml**—(Operational mode only) Display the command output as Junos XML protocol (Extensible Markup Language [XML]) tags.

except *pattern*—Ignore text matching a regular expression when searching the output. If the regular expression contains spaces, operators, or wildcard characters, enclose it in quotation marks.

find *pattern*—Display the output starting at the first occurrence of text matching a regular expression. If the regular expression contains spaces, operators, or wildcard characters, enclose it in quotation marks (" ").

hold—Hold text without exiting the **--More--** prompt.

Copyright © 2014, Juniper Networks, Inc.

last *lines*—Display the last number of lines you want to view from the end of the configuration. However, when the number of lines requested is less than the number of lines that the screen length setting permits you to display, Junos returns as many lines as permitted by the screen length setting. For more information on using the **last *lines*** option, see "Displaying Output Beginning with the Last Entries" on page 180.

match *pattern*—Search for text matching a regular expression. If the regular expression contains spaces, operators, or wildcard characters, enclose it in quotation marks.

no-more—Display output all at once rather than one screen at a time.

refresh *interval*—Refresh the display of the command according to the interval specified. The screen gets refreshed periodically to show you the current output of the command until you quit the command. The default refresh interval is one second. However, you can also explicitly specify a value from **1** through **604800** for the refresh interval.

request message (all | *account*@*terminal*)—Display command output on the terminal of a specific user logged in to your router, or on the terminals of all users logged in to your router.

resolve—(Operational mode only) Convert IP addresses into Domain Name System (DNS) names. Truncates to fit original size unless **full-names is** specified. To prevent the names from being truncated, use the **full-names** option.

save *filename*—Save the output to a file or URL. For information about specifying the filename, see "Specifying Filenames and URLs" on page 162.

tee—Display both the command output on screen and write it to a file. Unlike the UNIX **tee** command, if the file cannot be opened, just an error message is displayed.

trim *columns* —Trim specified number of columns from the start line.

Required Privilege Level

view

Related Documentation

- Displaying the Current Junos OS Configuration on page 87.
- Using the Pipe (|) Symbol to Filter Junos OS Command Output on page 173
- Using Regular Expressions with the Pipe (|) Symbol to Filter Junos OS Command Output on page 174
- Pipe (|) Filter Functions in the Junos OS Command-Line Interface on page 176

Copyright © 2014, Juniper Networks, Inc.

request

Syntax request <chassis | ipsec switch | message | mpls | routing-engine | security | services | system | flow-collector | support information>

Release Information Command introduced before Junos OS Release 7.4.

Description Stop or reboot router components, switch between primary and backup components, display messages, and display system information.

CAUTION: Halt the backup Routing Engine before you remove it or shut off the power to the router; otherwise, you might need to reinstall the Junos OS.

NOTE: If your router contains two Routing Engines and you want to shut the power off to the router or remove a Routing Engine, you must first halt the backup Routing Engine (if it has been upgraded) and then the master Routing Engine. To halt a Routing Engine, enter the **request system halt** command. You can also halt both Routing Engines at the same time by issuing the **request system halt both-routing-engines** command.

If you want to reboot a router that has two Routing Engines, reboot the backup Routing Engine (if you have upgraded it) and then the master Routing Engine.

NOTE: If you reboot the TX Matrix router, all the T640 master Routing Engines connected to the TX Matrix router reboot. If you halt both Routing Engines on a TX Matrix router, all the T640 Routing Engines connected to the TX Matrix router are also halted. Likewise, if you reboot the TX Matrix Plus router, all the T1600 or T4000 master Routing Engines connected to the TX Matrix Plus router reboot. If you halt both Routing Engines on a TX Matrix Plus router, all the T1600 or T4000 Routing Engines connected to the TX Matrix Plus router are also halted.

NOTE: If you insert a Flexible PIC Concentrator (FPC) into your router, you may need to issue the **request chassis fpc** command (or press the online button) to bring the FPC online. This applies to FPCs in M20, M40, M40e, M160, M320, and T Series routers. For command usage, see the **request chassis fpc** command description in the CLI Explorer.

Copyright © 2014, Juniper Networks, Inc.

Additional Information Most **request** commands are described in the *Junos System Basics and Services Command Reference*. The following **request** commands are described in the *Junos Interfaces Command Reference*: **request ipsec switch** and **request services**.

Required Privilege Level maintenance

Related Documentation
- Overview of Junos OS CLI Operational Mode Commands on page 149

Copyright © 2014, Juniper Networks, Inc.

request system commit server pause

Syntax **request system commit server pause**

Release Information Command introduced in Junos OS Release 12.1.

Description Pause the commit server.

NOTE: If you issue this command when a commit job is in process, the batch commit server pauses only after the current commit job is completed.

Options This command has no options.

Required Privilege Level view

Related Documentation

- Example: Configuring Batch Commit Server Properties on page 107

Sample Output

When you enter the **request system commit server pause** command, you are provided feedback on the status of your request.

request system commit server pause

```
user@host> request system commit server pause

Successfully paused the commit server.
```

Copyright © 2014, Juniper Networks, Inc.

request system commit server queue cleanup

Syntax **request system commit server queue cleanup**
<id *commit-id*>
<job-status (error| pending| success)>

Release Information Command introduced in Junos OS Release 12.1.

Description Clean up the batch commit queue.

Options **id *commit-id***—(Optional) Clean up batch commit operation status messages for a specific commit ID.

job-status—(Optional) Clean up batch commit operation status messages for the following:

- **error**—Clean up status messages for batch commit operations that have errors.
- **pending**—Clean up status messages for batch commit operations that are pending.
- **success**—Clean up status messages for batch commit operations that are successful.

Required Privilege Level view

Related Documentation
- Example: Configuring Batch Commit Server Properties on page 107

List of Sample Output
request system commit server queue cleanup id on page 314
request system commit server queue cleanup job-status success on page 314

Sample Output

When you enter the **request system commit server queue cleanup** command, you are provided feedback on the status of your request.

request system commit server queue cleanup id

```
user@host> request system commit server queue cleanup id 1008

Successfully cleaned up jobs.
```

request system commit server queue cleanup job-status success

```
user@host> request system commit server queue cleanup job-status success

Successfully cleaned up jobs.
```


Copyright © 2014, Juniper Networks, Inc.

request system commit server start

Syntax **request system commit server start**

Release Information Command introduced in Junos OS Release 12.1.

Description Start the commit server.

Options This command has no options.

Required Privilege Level view

Related Documentation
- Example: Configuring Batch Commit Server Properties on page 107

Sample Output

When you enter the **request system commit server start** command, you are provided feedback on the status of your request.

request system commit server start

```
user@host> request system commit server start

Successfully started the commit server.
```

restart

List of Syntax

Syntax on page 316
Syntax (ACX Series Routers) on page 316
Syntax (EX Series Switches) on page 316
Syntax (Routing Matrix) on page 317
Syntax (J Series Routing Platform) on page 317
Syntax (TX Matrix Routers) on page 317
Syntax (TX Matrix Plus Routers) on page 317
Syntax (MX Series Routers) on page 317
Syntax (J Series Routers) on page 318
Syntax (QFX Series) on page 318

Syntax

```
restart
<adaptive-services |ancpd-service | application-identification |audit-process |
  auto-configuration |captive-portal-content-delivery |ce-l2tp-service |chassis-control |
  class-of-service |clksyncd-service |database-replication|datapath-trace-service
  |dhcp-service | diameter-service | disk-monitoring | dynamic-flow-capture |
  ecc-error-logging | ethernet-connectivity-fault-management
  |ethernet-link-fault-management |event-processing | firewall
  |general-authentication-service | gracefully | iccp-service |idp-policy | immediately
  |interface-control | ipsec-key-management | kernel-replication | l2-learning | l2cpd-service
  | l2tp-service | l2tp-universal-edge | lacp | license-service |link-management
  |local-policy-decision-function |mac-validation |mib-process | mobile-ip | mountd-service
  |mpls-traceroute |mspd | multicast-snooping |named-service | nfsd-service |
  packet-triggered-subscribers |peer-selection-service |pgcp-service | pgm |
  pic-services-logging | pki-service |ppp | ppp-service |pppoe |
  protected-system-domain-service | redundancy-interface-process | remote-operations |
  root-system-domain-service | routing <logical-system logical-system-name>  | sampling
  | sbc-configuration-process | sdk-service |service-deployment | services | services pgcp
  gateway gateway-name | snmp |soft |static-subscribers |statistics-service|
  subscriber-management | subscriber-management-helper | tunnel-oamd |usb-control|
  vrrp |web-management>
<gracefully | immediately | soft>
```

Syntax (ACX Series Routers)

```
restart
<adaptive-services |audit-process | auto-configuration | autoinstallation |chassis-control |
  class-of-service |clksyncd-service |database-replication| dhcp-service | diameter-service
  | disk-monitoring | dynamic-flow-capture | ethernet-connectivity-fault-management
  |ethernet-link-fault-management |event-processing | firewall
  |general-authentication-service | gracefully | immediately |interface-control |
  ipsec-key-management | l2-learning | lacp |link-management |mib-process | mobile-ip |
  mountd-service |mpls-traceroute |mspd | named-service | nfsd-service | pgm | pki-service
  |ppp | pppoe | redundancy-interface-process | remote-operations | routing | sampling |
  sdk-service |secure-neighbor-discovery | service-deployment | services | snmp |soft
  |statistics-service| subscriber-management | subscriber-management-helper | tunnel-oamd
  | vrrp>
```

Syntax (EX Series Switches)

```
restart
<autoinstallation | chassis-control | class-of-service | database-replication | dhcp |
  dhcp-service | diameter-service | dot1x-protocol | ethernet-link-fault-management |
  ethernet-switching | event-processing | firewall | general-authentication-service |
  interface-control | kernel-replication | l2-learning | lacp | license-service | link-management
```


Copyright © 2014, Juniper Networks, Inc.

| lldpd-service | mib-process | mountd-service | multicast-snooping | pgm | redundancy-interface-process | remote-operations | routing | secure-neighbor-discovery | service-deployment | sflow-service | snmp | vrrp | web-management>

Syntax (Routing Matrix)

restart
<adaptive-services | audit-process | chassis-control | class-of-service | disk-monitoring | dynamic-flow-capture | ecc-error-logging | event-processing | firewall | interface-control | ipsec-key-management | kernel-replication | l2-learning | l2tp-service | lacp | link-management | mib-process | pgm | pic-services-logging | ppp | pppoe | redundancy-interface-process | remote-operations | routing <logical-system *logical-system-name*> | sampling | service-deployment | snmp>
<all | all-lcc | lcc *number*>
<gracefully | immediately | soft>

Syntax (J Series Routing Platform)

restart
<adaptive-services | audit-process | chassis-control | class-of-service | dhcp | dialer-services | dlsw | event-processing | firewall | interface-control | ipsec-key-management | isdn-signaling | l2-learning | l2tp-service | mib-process | network-access-service | pgm | ppp | pppoe | remote-operations | routing <logical-system *logical-system-name*> | sampling | service-deployment | snmp | usb-control | web-management>
<gracefully | immediately | soft>

Syntax (TX Matrix Routers)

restart
<adaptive-services | audit-process | chassis-control | class-of-service | dhcp-service | diameter-service | disk-monitoring | dynamic-flow-capture | ecc-error-logging | event-processing | firewall | interface-control | ipsec-key-management | kernel-replication | l2-learning | l2tp-service | lacp | link-management | mib-process |pgm | pic-services-logging | ppp | pppoe | redundancy-interface-process | remote-operations | routing <logical-system *logical-system-name*> | sampling | service-deployment | snmp| statistics-service>
<all-chassis | all-lcc | lcc *number* | scc>
<gracefully | immediately | soft>

Syntax (TX Matrix Plus Routers)

restart
<adaptive-services | audit-process | chassis-control | class-of-service | dhcp-service | diameter-service | disk-monitoring | dynamic-flow-capture | ecc-error-logging | event-processing | firewall | interface-control | ipsec-key-management | kernel-replication | l2-learning | l2tp-service | lacp | link-management | mib-process | pgm | pic-services-logging | ppp | pppoe | redundancy-interface-process | remote-operations | routing <logical-system *logical-system-name*> | sampling | service-deployment | snmp| statistics-service>
<all-chassis | all-lcc | all-sfc | lcc *number* | sfc *number*>
<gracefully | immediately | soft>

Syntax (MX Series Routers)

restart
<adaptive-services | ancpd-service |application-identification|audit-process | auto-configuration |captive-portal-content-delivery |ce-l2tp-service |chassis-control | class-of-service |clksyncd-service |database-replication| datapath-trace-service |dhcp-service | diameter-service | disk-monitoring | dynamic-flow-capture | ecc-error-logging | ethernet-connectivity-fault-management |ethernet-link-fault-management |event-processing | firewall | general-authentication-service | gracefully |iccp-service |idp-policy |immediately |interface-control | ipsec-key-management | kernel-replication | l2-learning |l2cpd-service |l2tp-service | l2tp-universal-edge | lacp | license-service |link-management |local-policy-decision-function |mac-validation |mib-process | mobile-ip |mountd-service |mpls-traceroute |mspd | multicast-snooping |named-service | nfsd-service |

packet-triggered-subscribers |peer-selection-service | pgcp-service | pgm |
pic-services-logging | pki-service |ppp | ppp-service |pppoe |
protected-system-domain-service | redundancy-interface-process | remote-operations
|root-system-domain-service | routing |routing <logical-system *logical-system-name*> |
sampling | sbc-configuration-process | sdk-service |service-deployment |services | services
pgcp gateway *gateway-name* |snmp |soft |static-subscribers |statistics-service|
subscriber-management | subscriber-management-helper | tunnel-oamd | usb-control|
vrrp |web-management>
<all-members>
<gracefully | immediately | soft>
<local>
<member *member-id*>

Syntax (J Series Routers)

restart
<adaptive-services | audit-process | chassis-control | class-of-service | dhcp | dhcp-service
| dialer-services | diameter-service | dlsw | event-processing | firewall | interface-control |
ipsec-key-management | isdn-signaling | l2ald | l2-learning | l2tp-service | mib-process |
network-access-service | pgm | ppp | pppoe | remote-operations | routing <logical-system
logical-system-name> | sampling | service-deployment | snmp | usb-control |
web-management>
<gracefully | immediately | soft>

Syntax (QFX Series)

restart
<adaptive-services | audit-process | chassis-control | class-of-service | dialer-services |
diameter-service | dlsw | ethernet-connectivity | event-processing | fibre-channel | firewall
| general-authentication-service | igmp-host-services | interface-control |
ipsec-key-management | isdn-signaling | l2ald | l2-learning | l2tp-service | mib-process |
named-service | network-access-service | nstrace-process | pgm | ppp | pppoe |
redundancy-interface-process | remote-operations |*logical-system-name*> | routing |
sampling |secure-neighbor-discovery | service-deployment | snmp | usb-control |
web-management>
<gracefully | immediately | soft>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.1 for the QFX Series.
Command introduced in Junos OS Release 12.2 for ACX Series routers.
Options added:

- **dynamic-flow-capture** in Junos OS Release 7.4.
- **dlsw** in Junos OS Release 7.5.
- **event-processing** in Junos OS Release 7.5.
- **ppp** in Junos OS Release 7.5.
- **l2ald** in Junos OS Release 8.0.
- **link-management** in Release 8.0.
- **pgcp-service** in Junos OS Release 8.4.
- **sbc-configuration-process** in Junos OS Release 9.5.
- **services pgcp gateway** in Junos OS Release 9.6.
- **sfc** and **all-sfc** for the TX Matrix Router in Junos OS Release 9.6.

Copyright © 2014, Juniper Networks, Inc.

Description Restart a Junos OS process.

CAUTION: Never restart a software process unless instructed to do so by a customer support engineer. A restart might cause the router or switch to drop calls and interrupt transmission, resulting in possible loss of data.

Options **none**—Same as **gracefully**.

adaptive-services—(Optional) Restart the configuration management process that manages the configuration for stateful firewall, Network Address Translation (NAT), intrusion detection services (IDS), and IP Security (IPsec) services on the Adaptive Services PIC.

all-chassis—(TX Matrix and TX Matrix Plus routers only) (Optional) Restart the software process on all chassis.

all-lcc—(TX Matrix and TX Matrix Plus routers only) (Optional) For a TX Matrix router, restart the software process on all T640 routers connected to the TX Matrix router. For a TX Matrix Plus router, restart the software process on all T1600 routers connected to the TX Matrix Plus router.

all-members—(MX Series routers only) (Optional) Restart the software process for all members of the Virtual Chassis configuration.

all-sfc—(TX Matrix Plus routers only) (Optional) For a TX Matrix Plus router, restart the software processes for the TX Matrix Plus router (or switch-fabric chassis).

ancpd-service—(Optional) Restart the Access Node Control Protocol (ANCP) process, which works with a special Internet Group Management Protocol (IGMP) session to collect outgoing interface mapping events in a scalable manner.

application-identification—(Optional) Restart the process that identifies an application using intrusion detection and prevention (IDP) to allow or deny traffic based on applications running on standard or nonstandard ports.

audit-process—(Optional) Restart the RADIUS accounting process that gathers statistical data that can be used for general network monitoring, analyzing, and tracking usage patterns, for billing a user based on the amount of time or type of services accessed.

auto-configuration—(Optional) Restart the Interface Auto-Configuration process.

autoinstallation—(EX Series switches only) (Optional) Restart the autoinstallation process.

captive-portal-content-delivery—(Optional) Restart the HTTP redirect service by specifying the location to which a subscriber's initial Web browser session is redirected, enabling initial provisioning and service selection for the subscriber.

Copyright © 2014, Juniper Networks, Inc.

ce-l2tp-service—(M10, M10i, M7i, and MX Series routers only) (Optional) Restart the Universal Edge Layer 2 Tunneling Protocol (L2TP) process, which establishes L2TP tunnels and Point-to-Point Protocol (PPP) sessions through L2TP tunnels.

chassis-control—(Optional) Restart the chassis management process.

class-of-service—(Optional) Restart the class-of-service (CoS) process, which controls the router's or switch's CoS configuration.

clksyncd-service—(Optional) Restart the external clock synchronization process, which uses synchronous Ethernet (SyncE).

database-replication—(EX Series switches and MX Series routers only) (Optional) Restart the database replication process.

datapath-trace-service—(Optional) Restart the packet path tracing process.

dhcp—(J Series routers and EX Series switches only) (Optional) Restart the software process for a Dynamic Host Configuration Protocol (DHCP) server. A DHCP server allocates network IP addresses and delivers configuration settings to client hosts without user intervention.

dhcp-service—(Optional) Restart the Dynamic Host Configuration Protocol process.

dialer-services—(J Series routers and EX Series switches only) (Optional) Restart the ISDN dial-out process.

diameter-service—(Optional) Restart the diameter process.

disk-monitoring—(Optional) Restart disk monitoring, which checks the health of the hard disk drive on the Routing Engine.

dlsw—(J Series routers and QFX Series only) (Optional) Restart the data link switching (DLSw) service.

dot1x-protocol—(EX Series switches only) (Optional) Restart the port-based network access control process.

dynamic-flow-capture—(Optional) Restart the dynamic flow capture (DFC) process, which controls DFC configurations on Monitoring Services III PICs.

ecc-error-logging—(Optional) Restart the error checking and correction (ECC) process, which logs ECC parity errors in memory on the Routing Engine.

ethernet-connectivity-fault-management—(Optional) Restart the process that provides IEEE 802.1ag Operation, Administration, and Management (OAM) connectivity fault management (CFM) database information for CFM maintenance association end points (MEPs) in a CFM session.

ethernet-link-fault-management—(EX Series switches and MX Series routers only) (Optional) Restart the process that provides the OAM link fault management (LFM) information for Ethernet interfaces.

Copyright © 2014, Juniper Networks, Inc.

ethernet-switching—(EX Series switches only) (Optional) Restart the Ethernet switching process.

event-processing—(Optional) Restart the event process (eventd).

fibre-channel—(QFX Series only) (Optional) Restart the Fibre Channel process.

firewall—(Optional) Restart the firewall management process, which manages the firewall configuration and enables accepting or rejecting packets that are transiting an interface on a router or switch.

general-authentication-service—(EX Series switches and MX Series routers only) (Optional) Restart the general authentication process.

gracefully—(Optional) Restart the software process.

iccp-service—(Optional) Restart the Inter-Chassis Communication Protocol (ICCP) process.

idp-policy—(Optional) Restart the intrusion detection and prevention (IDP) protocol process.

immediately—(Optional) Immediately restart the software process.

interface-control—(Optional) Restart the interface process, which controls the router's or switch's physical interface devices and logical interfaces.

ipsec-key-management—(Optional) Restart the IPsec key management process.

isdn-signaling—(J Series routers and QFX Series only) (Optional) Restart the ISDN signaling process, which initiates ISDN connections.

kernel-replication—(Optional) Restart the kernel replication process, which replicates the state of the backup Routing Engine when graceful Routing Engine switchover (GRES) is configured.

l2-learning—(Optional) Restart the Layer 2 address flooding and learning process.

l2cpd-service—(Optional) Restart the Layer 2 Control Protocol process, which enables features such as Layer 2 protocol tunneling and nonstop bridging.

l2tp-service— (M10, M10i, M7i, and MX Series routers only) (Optional) Restart the Layer 2 Tunneling Protocol (L2TP) process, which sets up client services for establishing Point-to-Point Protocol (PPP) tunnels across a network and negotiating Multilink PPP if it is implemented.

l2tp-universal-edge—(MX Series routers only) (Optional) Restart the L2TP process, which establishes L2TP tunnels and PPP sessions through L2TP tunnels.

lacp—(Optional) Restart the Link Aggregation Control Protocol (LACP) process. LACP provides a standardized means for exchanging information between partner systems on a link to allow their link aggregation control instances to reach agreement on the identity of the LAG to which the link belongs, and then to move the link to that LAG,

Copyright © 2014, Juniper Networks, Inc.

and to enable the transmission and reception processes for the link to function in an orderly manner.

lcc ***number***—(TX Matrix and TX Matrix Plus routers only) (Optional) For a TX Matrix router, restart the software process for a specific T640 router that is connected to the TX Matrix router. For a TX Matrix Plus router, restart the software process for a specific router that is connected to the TX Matrix Plus router.

Replace *number* with the following values depending on the LCC configuration:

- 0 through 3, when T640 routers are connected to a TX Matrix router in a routing matrix.
- 0 through 3, when T1600 routers are connected to a TX Matrix Plus router in a routing matrix.
- 0 through 7, when T1600 routers are connected to a TX Matrix Plus router with 3D SIBs in a routing matrix.
- 0, 2, 4, or 6, when T4000 routers are connected to a TX Matrix Plus router with 3D SIBs in a routing matrix.

license-service—(EX Series switches only) (Optional) Restart the feature license management process.

link-management— (TX Matrix and TX Matrix Plus routers and EX Series switches only) (Optional) Restart the Link Management Protocol (LMP) process, which establishes and maintains LMP control channels.

lldpd-service—(EX Series switches only) (Optional) Restart the Link Layer Discovery Protocol (LLDP) process.

local—(MX Series routers only) (Optional) Restart the software process for the local Virtual Chassis member.

local-policy-decision-function— (Optional) Restart the process for the Local Policy Decision Function, which regulates collection of statistics related to applications and application groups and tracking of information about dynamic subscribers and static interfaces.

mac-validation— (Optional) Restart the Media Access Control (MAC) validation process, which configures MAC address validation for subscriber interfaces created on demux interfaces in dynamic profiles on MX Series routers.

member ***member-id***—(MX Series routers only) (Optional) Restart the software process for a specific member of the Virtual Chassis configuration. Replace ***member-id*** with a value of **0** or **1**.

mib-process—(Optional) Restart the Management Information Base (MIB) version II process, which provides the router's MIB II agent.

mobile-ip—(Optional) Restart the Mobile IP process, which configures Junos OS Mobile IP features.

Copyright © 2014, Juniper Networks, Inc.

mountd-service—(EX Series switches and MX Series routers only) (Optional) Restart the service for NFS mount requests.

mpls-traceroute—(Optional) Restart the MPLS Periodic Traceroute process.

mspd—(Optional) Restart the Multiservice process.

multicast-snooping—(EX Series switches and MX Series routers only) (Optional) Restart the multicast snooping process, which makes Layer 2 devices, such as VLAN switches, aware of Layer 3 information, such as the media access control (MAC) addresses of members of a multicast group.

named-service—(Optional) Restart the DNS Server process, which is used by a router or a switch to resolve hostnames into addresses.

network-access-service—(J Series routers and QFX Series only) (Optional) Restart the network access process, which provides the router's Challenge Handshake Authentication Protocol (CHAP) authentication service.

nfsd-service—(Optional) Restart the Remote NFS Server process, which provides remote file access for applications that need NFS-based transport.

packet-triggered-subscribers—(Optional) Restart the packet-triggered subscribers and policy control (PTSP) process, which allows the application of policies to dynamic subscribers that are controlled by a subscriber termination device.

peer-selection-service—(Optional) Restart the Peer Selection Service process.

pgcp-service—(Optional) Restart the pgcpd service process running on the Routing Engine. This option does not restart pgcpd processes running on mobile station PICs. To restart pgcpd processes running on mobile station PICs, use the **services pgcp gateway** option.

pgm—(Optional) Restart the process that implements the Pragmatic General Multicast (PGM) protocol for assisting in the reliable delivery of multicast packets.

pic-services-logging—(Optional) Restart the logging process for some PICs. With this process, also known as fsad (the file system access daemon), PICs send special logging information to the Routing Engine for archiving on the hard disk.

pki-service—(Optional) Restart the PKI Service process.

ppp—(Optional) Restart the Point-to-Point Protocol (PPP) process, which is the encapsulation protocol process for transporting IP traffic across point-to-point links.

ppp-service—(Optional) Restart the Universal Edge PPP process, which is the encapsulation protocol process for transporting IP traffic across Universal Edge routers.

pppoe—(Optional) Restart the Point-to-Point Protocol over Ethernet (PPPoE) process, which combines PPP that typically runs over broadband connections with the Ethernet link-layer protocol that allows users to connect to a network of hosts over a bridge or access concentrator.

Copyright © 2014, Juniper Networks, Inc.

protected-system-domain-service—(Optional) Restart the Protected System Domain (PSD) process.

redundancy-interface-process—(Optional) Restart the ASP redundancy process.

remote-operations—(Optional) Restart the remote operations process, which provides the ping and traceroute MIBs.

root-system-domain-service—(Optional) Restart the Root System Domain (RSD) service.

routing—(ACX Series routers, QFX Series, EX Series switches, and MX Series routers only) (Optional) Restart the routing protocol process.

routing <logical-system *logical-system-name*>—(Optional) Restart the routing protocol process, which controls the routing protocols that run on the router or switch and maintains the routing tables. Optionally, restart the routing protocol process for the specified logical system only.

sampling—(Optional) Restart the sampling process, which performs packet sampling based on particular input interfaces and various fields in the packet header.

sbc-configuration-process—(Optional) Restart the session border controller (SBC) process of the border signaling gateway (BSG).

scc—(TX Matrix routers only) (Optional) Restart the software process on the TX Matrix router (or switch-card chassis).

sdk-service—(Optional) Restart the SDK Service process, which runs on the Routing Engine and is responsible for communications between the SDK application and Junos OS. Although the SDK Service process is present on the router, it is turned off by default.

secure-neighbor-discovery—(QFX Series, EX Series switches, and MX Series routers only) (Optional) Restart the secure Neighbor Discovery Protocol (NDP) process, which provides support for protecting NDP messages.

sfc *number*—(TX Matrix Plus routers only) (Optional) Restart the software process on the TX Matrix Plus router (or switch-fabric chassis). Replace ***number*** with **0**.

service-deployment—(Optional) Restart the service deployment process, which enables Junos OS to work with the Session and Resource Control (SRC) software.

services—(Optional) Restart a service.

services pgcp gateway *gateway-name*—(Optional) Restart the pgcpd process for a specific border gateway function (BGF) running on an MS-PIC. This option does not restart the pgcpd process running on the Routing Engine. To restart the pgcpd process on the Routing Engine, use the **pgcp-service** option.

sflow-service—(EX Series switches only) (Optional) Restart the flow sampling (sFlow technology) process.

Copyright © 2014, Juniper Networks, Inc.

snmp—(Optional) Restart the SNMP process, which enables the monitoring of network devices from a central location and provides the router's or switch's SNMP master agent.

soft—(Optional) Reread and reactivate the configuration without completely restarting the software processes. For example, BGP peers stay up and the routing table stays constant. Omitting this option results in a graceful restart of the software process.

static-subscribers—(Optional) Restart the static subscribers process, which associates subscribers with statically configured interfaces and provides dynamic service activation and activation for these subscribers.

statistics-service—(Optional) Restart the process that manages the Packet Forwarding Engine statistics.

subscriber-management—(Optional) Restart the Subscriber Management process.

subscriber-management-helper—(Optional) Restart the Subscriber Management Helper process.

tunnel-oamd—(Optional) Restart the Tunnel OAM process, which enables the Operations, Administration, and Maintenance of Layer 2 tunneled networks. Layer 2 protocol tunneling (L2PT) allows service providers to send Layer 2 PDUs across the provider's cloud and deliver them to Juniper Networks EX Series Ethernet Switches that are not part of the local broadcast domain.

usb-control—(J Series routers and MX Series routers only) (Optional) Restart the USB control process.

vrrp—(ACX Series routers, EX Series switches, and MX Series routers only) (Optional) Restart the Virtual Router Redundancy Protocol (VRRP) process, which enables hosts on a LAN to make use of redundant routing platforms on that LAN without requiring more than the static configuration of a single default route on the hosts.

web-management—(J Series routers, QFX Series, EX Series switches, and MX Series routers only) (Optional) Restart the Web management process.

Required Privilege Level reset

Related Documentation
- Overview of Junos OS CLI Operational Mode Commands on page 149

List of Sample Output restart interfaces on page 325

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

restart interfaces

```
user@host> restart interfaces
interfaces process terminated
interfaces process restarted
```

Copyright © 2014, Juniper Networks, Inc.

Copyright © 2014, Juniper Networks, Inc.

set

Syntax set <*statement-path*> *identifier*

Release Information Command introduced before Junos OS Release 7.4.

Description Create a statement hierarchy and set identifier values. This is similar to **edit** except that your current level in the hierarchy does not change.

Options *identifier*—Name of the statement or identifier to set.

statement-path—(Optional) Path to an existing statement hierarchy level. If that hierarchy level does not exist, it is created.

Required Privilege Level configure—To enter configuration mode, but other required privilege levels depend on where the statement is located in the configuration hierarchy.

Related Documentation

- edit on page 249
- Displaying the Current Junos OS Configuration on page 87

Copyright © 2014, Juniper Networks, Inc.

show system commit server queue

Syntax **show system commit server queue**
<id *commit-id*>
<job-status (all| error| pending| success)>
<patch (none | id *commit-id*) | (job-status (all | error | pending | success)>

Release Information Command introduced in Junos OS Release 12.1.

Description Display the status of commit server queue transactions.

NOTE: Only 50 successful commit jobs are stored in the database and displayed in the output. When the fifty-first job is committed, the first job is deleted from the database and is no longer displayed in the output.

Options **id *commit-id***—(Optional) Display the batch commit operation status messages for a specific commit ID.

job-status—(Optional) Display batch commit operation status messages for the following batch commit statuses:

- **all**—Status messages for all batch commit operations.
- **error**—Status messages for batch commit operations that have errors.
- **pending**—Status messages for batch commit operations that are pending.
- **success**—Status messages for batch commit operations that are successful.

patch (none | id *commit-id*) | job-status (all |error | pending | success)—(Optional) Display the patch file containing the configuration changes for all batch commit operations, a specific batch commit ID, or a specific job status.

Required Privilege Level view

Related Documentation
- Example: Configuring Batch Commit Server Properties on page 107

List of Sample Output
show system commit server queue on page 328
show system commit server queue job-status success on page 329
show system commit server queue patch on page 329

Sample Output

show system commit server queue

```
user@host> show system commit server queue

Pending commits:
  none
```


Copyright © 2014, Juniper Networks, Inc.

```
Completed commits:
  Id: 1000
  Last Modified: Tue Nov  1 22:46:43 2011
  Status: Successfully committed 1000

  Id: 1002
  Last Modified: Tue Nov  1 22:50:35 2011
  Status: Successfully committed 1002

  Id: 1004
  Last Modified: Tue Nov  1 22:51:48 2011
  Status: Successfully committed 1004

  Id: 1007
  Last Modified: Wed Nov  2 01:08:04 2011
  Status: Successfully committed 1007

  Id: 1009
  Last Modified: Wed Nov  2 01:16:45 2011
  Status: Successfully committed 1009

  Id: 1010
  Last Modified: Wed Nov  2 01:19:25 2011
  Status: Successfully committed 1010

  Id: 1011
  Last Modified: Wed Nov  2 01:28:16 2011
  Status: Successfully committed 1011


Error commits:
  Id: 1008
  Last Modified: Wed Nov  2 01:08:18 2011
  Status: Error while commiting 1008
```

show system commit server queue job-status success

```
user@host> show system commit server queue job-status success
Completed commits:
  Id: 1000
  Last Modified: Tue Nov  1 22:46:43 2011
  Status: Successfully committed 1000

  Id: 1001
  Last Modified: Tue Nov  1 22:47:02 2011
  Status: Successfully committed 1001
```

show system commit server queue patch

```
user@host> show system commit server queue patch
Pending commits:
  none


Completed commits:
  Id: 1000
  Last Modified: Tue Nov  1 22:46:43 2011
  Status: Successfully committed 1000

Patch:
[edit system commit]
```

```
+ server {
+     days-to-keep-error-logs 4294967295;
+     traceoptions {
+         file commitd_nov;
+         flag all;
+     }
+ }
  Id: 1002
  Last Modified: Tue Nov  1 22:50:35 2011
  Status: Successfully committed 1002

Patch:
[edit system commit server]
- days-to-keep-error-logs 4294967295;
  Id: 1004
  Last Modified: Tue Nov  1 22:51:48 2011
  Status: Successfully committed 1004

Patch:
[edit system commit server]
+ days-to-keep-error-logs 4294967295;
  Id: 1007
  Last Modified: Wed Nov  2 01:08:04 2011
  Status: Successfully committed 1007

Patch:
[edit system commit server]
-  days-to-keep-error-logs 4294967295;
+  days-to-keep-error-logs 2;
  Id: 1009
  Last Modified: Wed Nov  2 01:16:45 2011
  Status: Successfully committed 1009

Patch:
[edit]
+ snmp {
+     community abc;
+ }
  Id: 1010
  Last Modified: Wed Nov  2 01:19:25 2011
  Status: Successfully committed 1010

Patch:
[edit system syslog]
  file test { ... }
+ file j {
+     any any;
+ }
  Id: 1011
  Last Modified: Wed Nov  2 01:28:16 2011
  Status: Successfully committed 1011


Error commits:
  Id: 1008
  Last Modified: Wed Nov  2 01:08:18 2011
  Status: Error while commiting 1008

Patch:
[edit system]
+ radius-server {
```


Copyright © 2014, Juniper Networks, Inc.

```
+     10.1.1.1 port 222;
+ }
```

Copyright © 2014, Juniper Networks, Inc.

show system commit server status

Syntax **show system commit server status**

Release Information Command introduced in Junos OS Release 12.1.

Description Display commit server status.

NOTE: By default, the status of the commit server is “Not running”. The commit server starts running only when a commit job is added to the batch.

Options This command has no options.

Required Privilege Level view

Related Documentation
- Example: Configuring Batch Commit Server Properties on page 107

List of Sample Output
show system commit server status (When Server Is Inactive) on page 332
show system commit server status (When Server Is Active) on page 332

Sample Output

show system commit server status (When Server Is Inactive)

```
user@host> show system commit server status
Commit server status : Not running
```

show system commit server status (When Server Is Active)

```
user@R0> show system commit server status

Commit server status : Running
Jobs in process:
  1369 1370 1371
```


Copyright © 2014, Juniper Networks, Inc.

CHAPTER 15

Index

- Index on page 335

Copyright © 2014, Juniper Networks, Inc.

Index

Symbols

!
    in interface names....188
" ", configuration group wildcards....206
#, comments in configuration statements....xix, 84
( ), in syntax descriptions....xix
*
    in interface names....187
    regular expression operator....188
    wildcard character....206
+
    in statement lists....70
    regular expression operator....188
. (period)
    regular expression operator....188
/* */, comment delimiters....84
< >, in syntax descriptions....xix
?
    regular expression operator....206
    wildcard....206
[ ], in configuration statements....xix
\
    in interface names....187
    wildcard characters....206
{ }, in configuration statements....xix
    specifying statements....126
| (pipe)....309
    command output....309
    in syntax descriptions....xix, 309
| (pipe), in syntax descriptions....xix, 309

A

access privilege levels
    entering configuration mode....64
activate command....238
    usage guidelines....59
activate statements and identifiers....82
active configuration....23, 25
addresses
    machine name....34
annotate command....59, 239
    usage guidelines....84
append command....309
apply-groups statement....236
    usage guidelines....201
apply-groups-except statement....237
authorization *See* permissions

B

batch commit
    usage guidelines....107
braces, in configuration statements....xix
brackets
    angle, in syntax descriptions....xix
    square, in configuration statements....xix

C

candidate configuration....23, 25
CLI
    breadcrumbs
        usage guidelines....233
    command completion....286
    command history....56
        displaying....302
    comparing configuration versions....118, 122
    configuration mode
        description....58
        navigation commands, table....27
    configuration-breadcrumbs statement....245
    current working directory
        displaying....301
        setting....287
    date
        setting....295
    editing command line....185
    idle timeout, setting....288
    keyboard sequences....186
    permissions, displaying....299
    prompt strings....230
    prompt, setting....289
    restart, after software upgrade....290
    screen length, setting....291
    screen width, setting....292
    settings, displaying....296, 298
    terminal type, setting....293
    timestamp....230
    timestamp, setting....294
    type checking....128
    users, monitoring....157

Copyright © 2014, Juniper Networks, Inc.

word history....56
working directory....230
command history
operational mode....56
command output
configuration details....89
configuration, comparing files....176
end of, displaying from....180
filtering
comparing configuration versions....118, 122
number of lines, counting....178
pagination, preventing....180
regular expressions
first match, displaying from....179
matching output, displaying....180
nonmatching output, ignoring....179
retaining....179
RPC, displaying....178
saving to a file....181
sending to users....181
XML format, displaying....178
command shell....23
commands
completion....53, 231
configure....231
filenames, specifying....162
help about....49
history....56
options....153
URLs, specifying....162
comments
adding to configuration file....84
comments, in configuration statements....xix
commit
persisit-groups-inheritance....222
server....265
commit and-quit command
usage guidelines....100
commit at command
usage guidelines....103
commit command....240
usage guidelines....59, 98
commit comment command
usage guidelines....105
commit confirmed command
usage guidelines....102
commit scripts....28
commit synchronize command....240
commit synchronize statement
usage guidelines....147
commit | display detail command
usage guidelines....104
commit-interval statement....245
committing configuration
and exiting configuration mode....100
basic....98
confirmation required....102
logging message about....105
monitoring....104
scheduling for later....103
synchronizing on Routing Engines....144
compare command....309
usage guidelines....118, 122
compare filter....176
completing partial command entry....286
compress-configuration-files statement
usage guidelines....135
compressing configuration files....135
configuration
activating....120
adding comments....84
candidate....23, 25
committing....98
and exiting configuration mode....100
confirmation required....102
logging message about....105
monitoring process....104
scheduling for later....103
synchronizing on Routing Engines....144
comparing with previous....118, 122
deleting
statements....71
displaying
current configuration....268
details....89
edit command, using....69
files *See* configuration files
global replacement....189
groups configuration groups *See* configuration groups
locking....67
merging current and new....130
modifying....69
previous, displaying....121
protecting....137
replacing....130
saving to file....124, 125

Copyright © 2014, Juniper Networks, Inc.

storage of previous....117
unprotecting....137
configuration files
compressing....135
filename, specifying....162
saving to files....124, 125
URL, specifying....162
configuration groups
applying....201
creating....199
inheritance model....198
inherited values....204
interface parameters....210, 212
nested groups....201
overview....198
peer entities....214
re0, re1 groups....199
regional configurations....215
sets of statements....209
wildcards....206, 217
configuration mode, CLI....70, 98
command completion....53
commands
activate....59
annotate....59
commit....59
copy....59
deactivate....59
delete....59
edit....59
exit....59
extension....59
help....59
insert....59
load....59
paste....60
quit....60
rollback....46, 60
run....60
save....60
set....60
show....60
status....60
top....60
up....60
update....60
configuration hierarchy, description....62
description....58
entering....64
exiting....65
global replacement....189
identifier, description....61
locking....67
statement
container....62
description....61
leaf....62
switching to operational mode....33
top level statements, interpreting....61
users editing configuration
displaying....94
multiple simultaneous users....101
configuration mode, entering....304
configuration statements
adding comments about....84
deleting....71
help about....51
inheriting from groups....209
overviews....69
structure and components....126
configure command....304
names and addresses....34
usage guidelines....64, 150
configure exclusive command
usage guidelines....67
container hierarchy *See* hierarchy
conventions
text and syntax....xviii
copy command....246
usage guidelines....59, 150
count command....309
count filter....178
curly braces, in configuration statements....xix
current working directory
displaying....301
setting....287
cursor, moving....186
customer support....xx
contacting JTAC....xx

D

data types, CLI....128
date
setting from CLI....295
days-to-keep-error-logs statement....246
deactivate command....247
usage guidelines....59

Copyright © 2014, Juniper Networks, Inc.

deactivate statements and identifiers
usage guidelines....82
default configuration group....224
delete command....248
usage guidelines....59, 71
directories
working, displaying....301
disable statement
usage guidelines....82
display detail command
usage guidelines....89
display inheritance command
usage guidelines....204
display set command
usage guidelines....92
display xml filter....178
documentation
comments on....xix

E
edit command....249
usage guidelines....59
editing command line....185
Emacs keyboard sequences....185
environment settings, CLI
command completion....231
displaying....231
example configuration....232
idle timeout....230
prompt string....230
screen dimensions....229, 231
software upgrade, restarting after....230
terminal type....230
timestamp....230
working directory....230
except command....309
except filter....179
exit command....250
from configuration mode....33
usage guidelines....59, 65
exit configuration-mode command....250
usage guidelines....65
extension command
usage guidelines....59

F
file command....306
usage guidelines....150, 159
filenames, specifying in commands....162
files
configuration, compressing....135
listing....160
saving command output to....181
saving configurations to files....124, 125
viewing....159
find command....309
find filter....179
font conventions....xviii
FreeBSD UNIX kernel....24

G
groups statement....251
usage guidelines....199
when....282

H
help apropos command
usage guidelines....50
help command....253, 308
usage guidelines....50, 59
help reference command
usage guidelines....50
help tip cli command
usage guidelines....53
history, CLI commands
displaying....302
operational mode....56
hold command....309
hold filter....179

I
identifiers
inserting in sequential lists....75
renaming....75
specifying....126
idle timeout
user, setting....288
values, CLI sessions....230
ignore filter....179
inheritance model, configuration groups....198
inherited values, configuration groups....204
insert command....254
usage guidelines....59, 75
interface
configuring a routing protocol....40
interface names
conventions....158

Copyright © 2014, Juniper Networks, Inc.

interfaces
media parameters....210, 212
issuing relative configuration commands....74

J
J-Web graphical user interface (GUI)....28
juniper-ais configuration group
usage guidelines....199
juniper.conf file, compressing....135
Junos XML management protocol....28
junos-defaults configuration group....275
displaying....224, 271, 275
Junos-FIPS software environment....28

K
keyboard sequences
editing command line....185

L
last command....309
last filter....180
load command....255
usage guidelines....59
load merge command
usage guidelines....130
load override command
usage guidelines....130
load set command
usage guidelines....131
locking configuration....67
logical interfaces
unit numbers....158

M
manuals
comments on....xix
match command....309
match filter....180
maximum-aggegate-pool statement....256
maximum-entries statement....257

N
names
wildcard 217
naming conventions, interface....158
nested configuration groups....201
no-compress-configuration-files statement
usage guidelines....135
no-more command....309, 310
no-more filter....180

O
operational mode, CLI
command history....56
switching to configuration mode....33
users, monitoring....157
word history....56

P
parentheses, in syntax descriptions....xix
partial command entry, completing....286
paste command
usage guidelines....60
peer entities....214
permissions, CLI, displaying....299
pipe (|)
command output, filtering....176, 309
processes
managing....165
restarting....316
programs
managing....165
prompt
setting to display in CLI....289
prompt strings
CLI....230
protect command....258
usage guidelines....137
protecting configuration
usage guidelines....137

Q
quit command....150, 259
usage guidelines....60

R
re0 configuration group....199
re1 configuration group....199
redrawing screen....186
regional configurations....215
regular expressions
first match, displaying from....179
matching output, displaying....180
nonmatching output, ignoring....179
relative option....131
rename command....260
usage guidelines....75
renaming identifiers....75

Copyright © 2014, Juniper Networks, Inc.

replace command....261
usage guidelines....189
replace option....130
request command....311
usage guidelines....150
request message filter....181
request system configuration rescue delete command....124, 135
request system configuration rescue save command....123, 135
request system halt command....169
request system logout pid pid_number command....67
request system reboot command....169
resolve command....309
restart command....316
usage guidelines....150
restart routing command....168
restarting
after software upgrade....230
software processes....316
rollback command....46, 262
usage guidelines....60
Routing Engines
synchronizing configuration....144
synchronizing scripts....144
RPC
displaying command output in....178
run command....263
usage guidelines....60

S

save command....264, 309
usage guidelines....60, 124, 125
screen
dimensions....229, 231
redrawing....186
screen length, setting....291
screen width, setting....292
scripts
synchronizing on Routing Engines....144
set cli complete-on-space command....286
usage guidelines....231
set cli directory command....287
usage guidelines....230
set cli idle-timeout command....288
usage guidelines....230
set cli prompt command....289
usage guidelines....230
set cli restart-on-upgrade command....290
usage guidelines....230
set cli screen-length command....291
usage guidelines....229, 231
set cli screen-width command....292
set cli terminal command....293
usage guidelines....230
set cli timestamp command....294
usage guidelines....230
set command....69
configuration mode....266, 327
usage guidelines....60
set date command....295
set option....131
show cli authorization command....299
show cli command....296, 298
usage guidelines....231
show cli directory command....301
show cli history command....302
usage guidelines....56
show command
configuration mode....267
usage guidelines....60
show configuration command....268
show groups junos-defaults command....275
usage guidelines....224
show system processes extensive command....166
output, table....167
show version command
Junos OS....164
show | display inheritance command....271
show | display inheritance defaults command
usage guidelines....224
show | display omit command....272
show | display set command....273
usage guidelines....92
show | display set relative....274
show | display set relative command....274
usage guidelines....92
ssh command
usage guidelines....150
status command....276
usage guidelines....60, 94
storing previous configurations....117
strings
help about....50
support, technical *See* technical support
symbol....181
syntax conventions....xviii

Copyright © 2014, Juniper Networks, Inc.

T

technical support
contacting JTAC....xx
tee command....309
telnet command
usage guidelines....150
terminal screen
length, setting....291
width, setting....292
terminal type....230
setting....293
timeout, user, setting....288
timestamp, CLI output, setting....294
top command....277
usage guidelines....60, 74
traceoptions statement....278
trim command....309
TX Matrix router
configuration groups....199
configuration groups example....203
type checking, CLI....128

U

UNIX operating system....23, 24
UNIX shell....25
unprotect command....279
usage guidelines....137
unprotecting configuration
usage guidelines....137
up command....280
usage guidelines....60, 74
update command....281
usage guidelines....60, 86
updating configure private configuration....86
upgrade, restarting after....230
upgrading software....230
URLs, specifying in commands....162
user accounts
configuration example....34
user timeout, setting....288
users
CLI permissions, displaying....299
editing configuration
displaying....94
multiple simultaneous users....101
of CLI, monitoring....157

W

wildcard characters....206
wildcard command....283
wildcard delete command
usage guidelines....193
wildcard names....217
wildcard range command
usage guidelines....78
word history
operational mode....56
working directory
current, displaying....301
current, setting....287

X

XML format
displaying command output in....178

Copyright © 2014, Juniper Networks, Inc.

Copyright © 2014, Juniper Networks, Inc.